# 3-D Scene Reconstruction from Multiple Photometric Images

Christopher J. Forne, B.E. (Hons. I)

A thesis presented for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering
at the
University of Canterbury,
Christchurch, New Zealand.

30 April 2007

# ABSTRACT

This thesis deals with the problem of three dimensional scene reconstruction from multiple camera images. This is a well established problem in computer vision and has been significantly researched. In recent years some excellent results have been achieved, however existing algorithms often fall short of many biological systems in terms of robustness and generality. The aim of this research was to develop improved algorithms for reconstructing 3D scenes, with a focus on accurate system modelling and correctly dealing with occlusions.

With scene reconstruction the objective is to infer scene parameters describing the 3D structure of the scene from the data given by camera images. This is an ill-posed inverse problem, where an exact solution cannot be guaranteed. The use of a statistical approach to deal with the scene reconstruction problem is introduced and the differences between maximum a priori (MAP) and minimum mean square estimate (MMSE) considered. It is discussed how traditional stereo matching can be performed using a volumetric scene model. An improved model describing the relationship between the camera data and a discrete model of the scene is presented. This highlights some of the common causes of modelling errors, enabling them to be dealt with objectively.

The problems posed by occlusions are considered. Using a greedy algorithm the scene is progressively reconstructed to account for visibility interactions between regions and the idea of a complete scene estimate is established. Some simple and improved techniques for reliably assigning opaque voxels are developed, making use of prior information. Problems with variations in the imaging convolution kernel between images motivate the development of a pixel dissimilarity measure.

Belief propagation is then applied to better utilise prior information and obtain an improved global optimum. A new volumetric factor graph model is presented which represents the joint probability distribution of the scene and imaging system. By utilising the structure of the local compatibility functions, an efficient procedure for updating the messages is detailed. To help convergence, a novel approach of accentuating beliefs is shown. Results demonstrate the validity of this approach, however the reconstruction error is similar or slightly higher than from the Greedy algorithm.

To simplify the volumetric model, a new approach to belief propagation is demonstrated by applying it to a dynamic model. This approach is developed as an alternative

to the full volumetric model because it is less memory and computationally intensive. Using a factor graph, a volumetric known visibility model is presented which ensures the scene is complete with respect to all the camera images. Dynamic updating is also applied to a simpler single depth-map model. Results show this approach is unsuitable for the volumetric known visibility model, however, improved results are obtained with the simple depth-map model.

# ACKNOWLEDGEMENTS

First and foremost I would like to thank my supervisor Professor Michael Hayes. Michael's easy going nature and helpful approach to problem solving have helped me along the way, as well as helping to proofread my work. Thankyou for sharing your skills and knowledge with me.

Thanks also to the Electrical and Computer Engineering Department at the University of Canterbury, and to all the good friends I made there. Thanks to the Electrical engineering soccer team which often distracted and entertained me in my lunch hour.

This work was supported by a Bright Future Scholarship from the Foundation of Research, Science and Technology, New Zealand. I also appreciate the financial support I have received from the Prime Minister's Sports Scholarship.

A final thanks to my Mum and Dad for providing me with the opportunities that have lead to the writing of this thesis, keeping me motivated and providing me with 'food deliveries' in the final weeks. And also to my girlfriend Emily, for helping with proofreading and inspiring me to finish my thesis so we can go and explore the world together.

# CONTENTS

# Chapter 1

## INTRODUCTION

The problem of reconstructing or estimating a three-dimensional scene from sensor data is fundamental to many fields of science and engineering. One approach is to combine the data from multiple camera images located at different spatial positions. Commonly referred to as stereo reconstruction, this is a well established problem in computer vision. Stereo reconstruction has many practical applications, including robot navigation, virtual reality, topographic mapping and object recognition. Although significant work has been done on this problem, existing algorithms often still fall short of matching the performance of many biological systems, such as the human visual system, in terms of robustness and generality. The aim of this research was to develop improved algorithms for reconstructing 3D scenes, with a focus on accurate system modelling and correctly dealing with occlusions.

The stereo reconstruction problem is fundamental to many aspects of image and vision computing, particularly with the widespread development of virtual reality computer games, machine vision, and advanced special effects in the film industry. The problem was first investigated in 1849, when Aimé Laussedat used terrestrial photographs for topographic map compilation. At about the same time, investigations into human or biological stereopsis began, following the invention of the stereoscope by Sir Charles Wheatstone. Since then a large amount of research has been undertaken, and the problem has become a rapidly evolving and exciting field of study.

Many techniques exist for estimating the three dimensional structure within a scene. Usually these are based on direct measurements of the time of flight or phase shift of propagating waves. For example, sonar uses sound propagation (primarily under water), radar uses electromagnetic waves (usually in the microwave region), and LIDAR (Light-Imaging Detection and Ranging) uses laser light. Photometric stereo (PS) uses several images of the same surface taken from the same viewpoint but under illumination from different directions, to estimate local surface orientation. Especially applicable in the medical field is tomography, a volume reconstruction technique using x-rays and MRI, which involves imaging by sections or slices.

This thesis focuses on 3D scene reconstruction from multiple photometric camera

images for a number of reasons. Firstly it is passive, meaning that light or other forms of energy are not emitted by the imaging device. This is important, since the scene under observation will not be affected. It is also useful when imaging distant objects, as a large energy source is not required. Stereo reconstruction is scanless, allowing the entire scene to be captured almost instantaneously. This is particularly important for real time applications where the parts of the scene move relative to the camera. This motion leads to distortion and other problems if a scanning based method is used. In addition the obtained depth information is dense and aligned with visual information in common image coordinates. This is useful for further image processing and 3D modelling. Reconstruction from camera images is also a very cost effective solution to this problem.

## 1.1 STEREO RECONSTRUCTION

The objective of the stereo reconstruction problem is to reconstruct a 3D model of the scene from multiple camera images. These images contain information about the scene which relates to the scene radiances and structure. By forming a relationship between the scene parameters and the image data, the camera data can be used to infer information about the scene. This relationship is based on standard optical principles, which describe the transfer of light from the scene to the camera sensors. With stereo reconstruction the objective is to infer the scene parameters from the given data. This is an inverse problem, since the scene is unknown. To demonstrate the relationship between the camera images and the scene consider the diagram shown in Fig. 1.1. This figure shows a collection of objects being imaged by two cameras and the observed images.

A wide variety of methods have been developed to deal with stereo reconstruction. Traditionally, image based methods have been used [Lane and Thacker 1996] where regions or points are matched between pairs of images. The 3D structure of each matched primitive can then be found by triangulation. This worked for simple scenes but did not make good use of the available information. In recent years, research has centred around applications in machine vision and computer graphics, where accurate and detailed models of complex scenes, containing many discontinuous surfaces and semi-occluded regions, are required. For such applications, traditional techniques are unsuitable, requiring new and improved approaches to deal with multiple cameras, multiple surfaces, and widely varying views. This has lead to the development of various global optimisation techniques [Sun et al. 2003, Kolmogorov et al. 2003, Meltzer et al. 2005]. It has also lead to the development of volumetric techniques, where the scene estimate is formed directly in 3D [Culbertson et al. 1999, Seitz and Dyer 1999, Harding et al. 2000]. This is preferable to the traditional approaches, since the relationship between scene parameters and image data can be accurately modelled. A detailed survey of volumet-

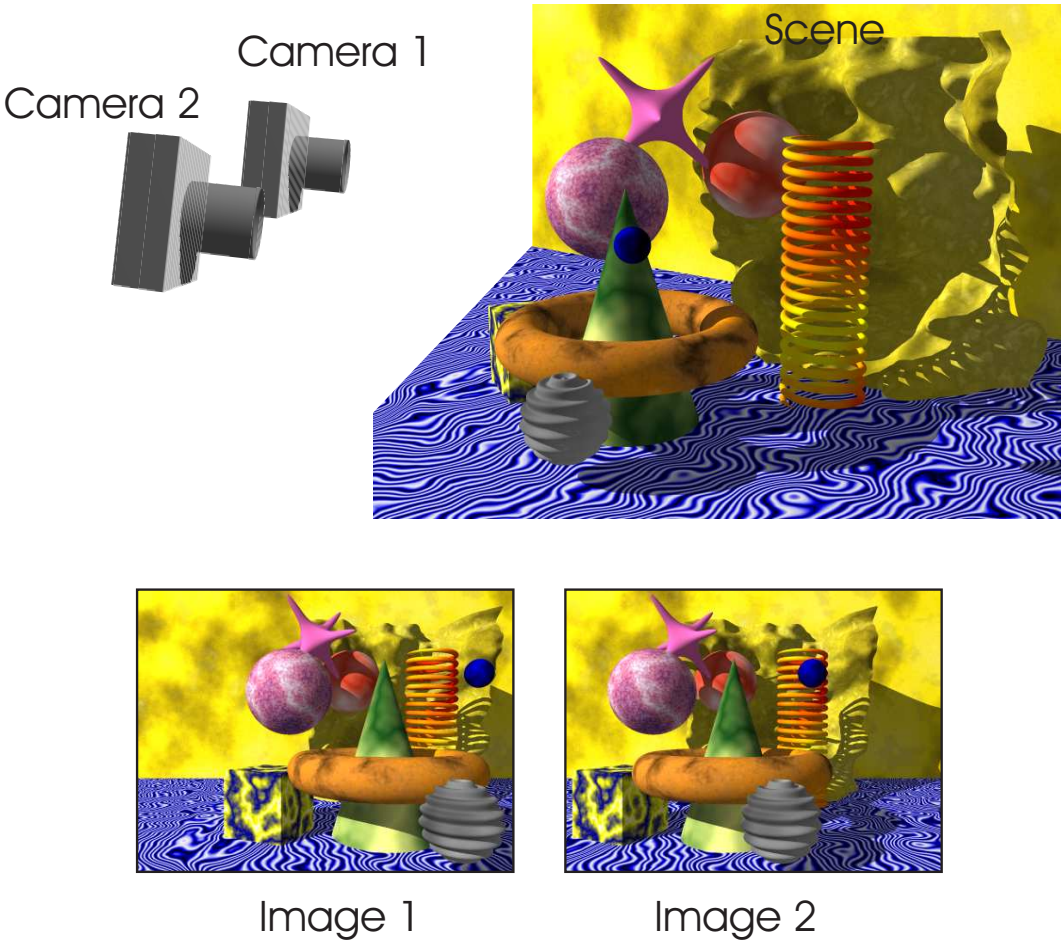


**Figure 1.1** Diagram of an imaging system showing two cameras observing a collection of objects and the observed images. The stereo reconstruction problem is to determine the structure and radiance of the scene from the observed images and any additional prior information.

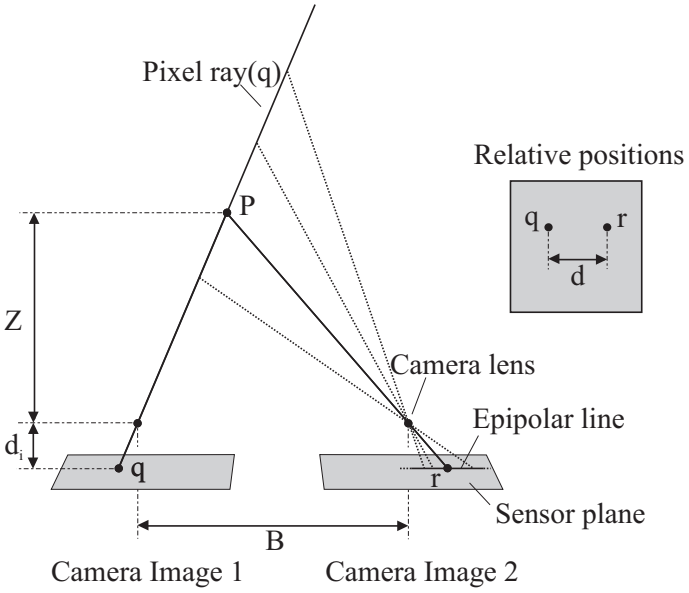


**Figure 1.2** Diagram of a typical stereo imaging system showing the camera geometry.

ric methods is given in Slabaugh et al. [2001]. A more recent survey is given in Seitz et al. [2006].

The traditional approach to stereo reconstruction involves matching points between two or more images to estimate the depth of objects within a scene. By viewing a 3D scene from different locations the position of a given point within each camera image will vary. As an example consider the point $P$ in Fig. 1.2. This has image coordinates $(x, y)$ as viewed from camera 1 and image coordinates $(x + d, y)$ when viewed from camera 2. By correctly matching this point between the two images the relative shift, or disparity, $d$, of the point can found. This can then be used to calculate the depth of the point. If all cameras have the same focal length, are parallel to each other, and located on the same plane, the magnitude of this disparity is related to the depth, $Z$, by

$$Z = \frac{Bd_i}{d}, \tag{1.1}$$

where $B$ is the distance between two cameras, commonly referred to as the baseline length, and $d_i$ is the distance of the image plane behind the principal point. For cameras focused at or near infinity this is approximately equal to focal length of the camera.

Like many tasks in image and vision computing, stereo reconstruction is an ill-posed problem with inherent ambiguities in the inverse solution. The process of projecting a 3D scene onto a 2D camera image results in an inherent information loss. The use of multiple cameras enables some 3D information to be regained. In addition to this loss of information there is other information lost through system noise, and an inability to model the image formation process accurately.

To deal with the ambiguities of stereo reconstruction prior information can be ap-

plied to improve the scene reconstruction. By adopting a statistical approach we can conceptualise stereo reconstruction as an estimation problem, where the objective is to form the most likely scene estimate given the camera data and any additional prior information. This is an optimisation problem over the joint probability distribution of the system. This is an extremely complex problem, consequently a wide variety of techniques for dealing with the stereo reconstruction problem have been developed. In this thesis the focus is on improved system models, and techniques for optimising these models.

## 1.2 NOVEL APPROACHES AND THESIS STRUCTURE

This thesis deals with the scene reconstruction problem using multiple camera images. Chapter 2 describes the image formation process and demonstrates an improved model for the stereo reconstruction problem, describing the relationship between the camera data and a discrete model of the scene. Chapter 3 provides an overview of reconstruction techniques, as well as background information about stereo reconstruction and discusses existing approaches to solving the problem.

Chapter 4 deals with the problems posed by occlusions using a greedy approach. In this approach the scene is progressively reconstructed to account for visibility interaction between regions. The term "visibility interaction" is used denote that the state of one region will affect the visibility of other regions. This extends on the work of Preddey and Lane [1997] and Harding et al. [2000], who formed a estimate of the scene by progressively building up a set of opaque surfaces. In this chapter the idea of a complete scene estimate is established. The reliability of the points selected at each iteration is improved by weighting the likelihood measure by the number of cameras which observe the point, as well as the inclusion of prior information.

Chapter 5 applies belief propagation to an improved volumetric model of the scene. Until recently most work on the scene reconstruction problem used a depth map model, or a variation of this, without focussing on the visibility interactions. Here a new model is presented which represents the joint probability distribution of the scene and imaging system. The local structure of the probability distribution within the model is utilised to compute the message updating more efficiently for this particular volumetric model. A simple technique for helping convergence is also described.

Chapter 6 presents a new approach to belief propagation by applying it to a changing statistical model. A dynamic model is adopted to simplify the full 3D system model, while still taking into account the visibility interaction between points. In the dynamic model scene visibilities are updated as the confidence in the scene estimate improves. Belief propagation is applied to this dynamic model to optimise the joint probability distribution of the system. The advantages of this model over the full 3D model presented in Chapter 5 are that it is less memory intensive and simpler to optimise, thus

making it much faster to compute. Results show this approach is unsuitable for the proposed volumetric known visibility model, however improved results are obtained with the simple depth-map model.

Finally, in Chapter 7 the conclusions and suggestions for future work are given.

The papers I have published during the course of my thesis are: [Bones et al. 2000, Forne et al. 2000, Forne and Hayes 2001, Forne and Hayes 2002, Forne and Hayes 2003, Barclay et al. 2003].

# Chapter 2

## SYSTEM MODELLING

To describe and estimate desired properties of a scene, such as structure and radiance, and relate these to the observed image data, a model or representation of the physical system is required. This model allows information about the scene to be derived or inferred from the image data and is a vital component of any scene reconstruction algorithm. The system model parameterises the scene and provides a mapping between the scene parameters and the image data. To be useful, this mapping must accurately relate the scene parameters to the image data in a well defined and usable fashion. The system model should also be as simple as possible to aid the reconstruction process. In addition, the model ought to include any additional prior information and relate this to the scene parameters.

The system model comprises three key components: the choice of scene parameters, a mapping between scene parameters and image data, and the incorporation of prior knowledge about the scene. The choice of scene parameters is significant, as the precise objective of scene reconstruction is defined in terms of these parameters. However, the other two components are equally important, and play a vital role in the scene reconstruction process. The mapping between scene parameters and image data, defines the relationship between these parameters, while the incorporation of prior knowledge provides additional information that can help make the scene estimate more accurate and reliable. Both of these components may contain uncertainties and should therefore be defined in probabilistic terms.

The three key components of the system model can be combined and represented using a single statistical model. This allows the system to be modelled in a well defined and cohesive way. By adopting a statistical model, uncertainties or soft constraints can be incorporated into the model along with exact or known quantities. With a statistical model the system is defined as a joint probability distribution over the set of scene parameters. This is usually expressed as the product of smaller sub-distributions in the form of a Bayesian Network, Markov Random Field, or Factor Graph. Given such a representation, the objective of scene reconstruction is to determine the state of these parameters or variables, so that the resulting scene estimate is optimal in some probabilistic sense.

Unfortunately, because of the complexity of the system model it is usually extremely difficult to find solutions that are optimal or even near optimal. To deal with this, simpler approximate models can be used, which although less accurate, are easier to optimise. This has lead to a wide variety of system models for scene reconstruction, each with its own advantages and disadvantages. Because of its strong influence on the resulting scene reconstruction, this choice of system model is extremely important.

## 2.1 SCENE PARAMETERS

A variety of parameters can be used to represent the radiometry of a three dimensional scene. The choice of these parameters is important, since it affects what properties can be estimated as well as how easy it is to estimate these properties. The scene parameters usually relate to the opacity and radiance of a scene, although reflection properties are sometimes modelled. Commonly parameters correspond to points or regions within the scene or images, however higher level parameters, such as the intensity edges or corners, can be used. It is also possible to use either discrete or continuous parameters, each having its own advantages and disadvantages. For applications such as virtual reality, animations and interactive visualisation, a detailed radiometric model of the scene is usually desired. This may include the reflective and transmissive properties of each surface as well as the location of various light sources. On the other hand for robotic applications, the location of visible surfaces may be all that is required. This leads to varying degrees of simplification.

Most scene models[1] are based on a detailed radiometric model of the physical environment. This model consists of a number of reflecting surfaces that are illuminated by one or more light sources, as shown in Fig. 2.1. The perceived brightness and colour of any surface point depends on the level and direction of light incident at that point as well as the surface reflection properties. These reflection properties can be accurately modelled using a Bidirectional Reflectance Distribution Function (BRDF) [Nicodemus et al. 1977]. The BRDF gives the ratio of reflected to incident light and is a function of incoming and outgoing angles as well surface position and wavelength. It is governed by the structural and optical properties of the surface and fully determines how a surface will appear under various lighting conditions. The advantage of such a reflectance model is that the surface model is independent of the scene lighting, making it consistent under changing illumination. This is useful in situations where there is a time delay between the capture of the images, such as with aerial photogrammetry, since the lighting conditions may change between images. Having a model of the reflectance is also extremely useful in computer graphic applications, where it is often necessary to render new views, under different lighting conditions or in the presence of additional objects.

[1]The term scene model is used throughout this thesis to defined a set of scene parameters and functions acting on these parameters, rather than a particular realisation of the scene parameters.

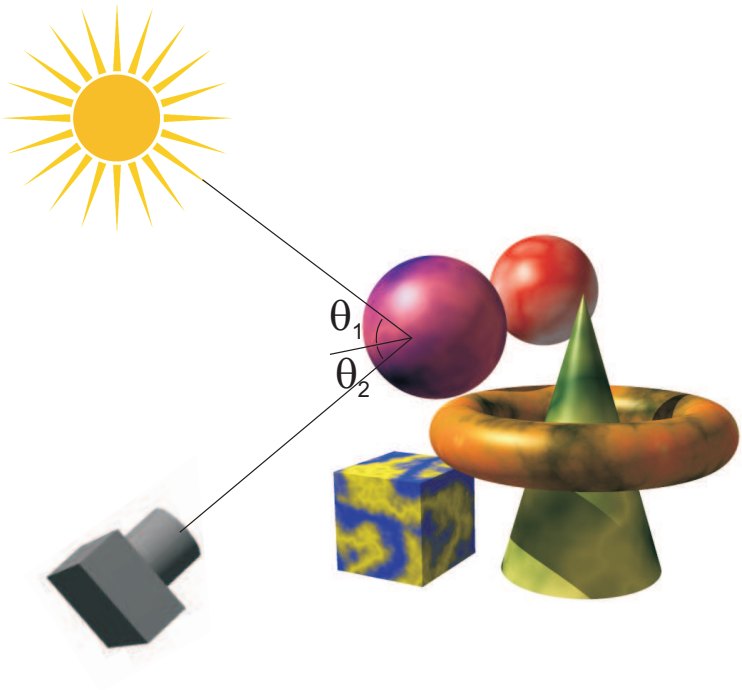


**Figure 2.1** Reflectance model of the scene. The perceived brightness and colour of any surface point depends on the reflectance properties of the surface as well as the level and direction of the incident light and observation direction.

The BRDF can be divided into three components: specular, directional diffuse, and uniform diffuse (commonly referred to as Lambertian). The specular component is the mirror-like component, where incident light from a single incoming direction is reflected onto a single outgoing direction at an equal and opposite angle with respect to the surface normal. At the other extreme is the Lambertian component, where incident light is reflected in all directions so that the radiance is uniformly distributed over a hemisphere surrounding the surface. This type of reflection has the rather useful property that a surface point will appear the same when viewed from all angles. Finally, there is the directional diffuse component, which is essentially any reflection that lies somewhere between specular and Lambertian.

In most cases a Lambertian reflectance model is assumed. This reduces the number of parameters required to represent the scene and simplifies the mapping between image and scene parameters, as all points will appear the same regardless of the viewing direction. For many scenes, especially natural ones with rough surfaces, this is a reasonable approximation so long as the lighting is fairly diffuse. In the ideal situation where a surface is illuminated evenly from all directions, there will be no apparent difference between specular and Lambertian reflections and so the Lambertian assumption will apply. Unfortunately, most scenes will contain some smooth shiny surfaces or areas that are illuminated by directional light sources. In this case the Lambertian assumption will be inaccurate, resulting in a poor system model. Fortunately, in many situations a

Lambertian model can still be used with good accuracy if the images are appropriately pre-filtered to remove most of the variation in observed intensity between images.

Given the reflectance of each surface and the position and radiance of all illuminating light sources, the reflected radiance from each point within the scene can be calculated. This, combined with the transmission properties of the scene, will determine the light incident upon the various cameras. Rather than modelling the various light sources and surface reflectance properties, a common approach is to simply model the radiance emitted from each point, whether this be reflected or generated. Although less general than the reflectance model, the resulting radiance model is good for most applications and is considerably simpler and easier to work with.

The scene transmittances can also be simplified by assuming that objects are either fully opaque or transparent. This is the standard approach taken with stereo reconstruction and results in binary transmittances. In most situations this assumption is approximately true, the notable exception being glass objects or windows, which both reflect and transmit light. Problems with opacity assumptions also occur at object boundaries where the average transmittance over a small region in the direction of a camera maybe somewhere between 0 and 1. Assuming that radiating and semi-opaque regions are reasonably this, the scene can be modelled as a set of surfaces within a three dimensional volume. The objective of stereo reconstruction then becomes the determination of the location and properties of these surfaces.

Scene models can be divided into three main groups depending on how they parameterise the various scene properties. The first approach is to represent the scene properties as a three dimensional volume. This is the most general and comprehensive approach, and the resulting scene models are referred to as volumetric models. The second approach is to represent the scene as a surface, defined in terms of its depth relative to a reference plane. Referred to as the depth-map model, this is the approach traditionally taken and requires object transmittances or opacities to be binary. The final approach is to express the scene in terms of objects or higher level features such as human, face, car, etc. Identifying and modelling such features is important in many applications, but does not provide a general purpose model. Therefore, these models will not be investigated in this thesis.

### 2.1.1 Volumetric models

Volumetric models are the most general and flexible way of modelling the scene. With this type of model the desired scene properties, such as radiance and opacity, are represented throughout a three dimensional volume as a function of spatial position (see Fig. 2.2(a)). Rather than parameterising opacity directly, the boundary between opaque and transparent regions is commonly modelled instead. These boundaries correspond to object surfaces, and are often simpler to model and relate with prior information

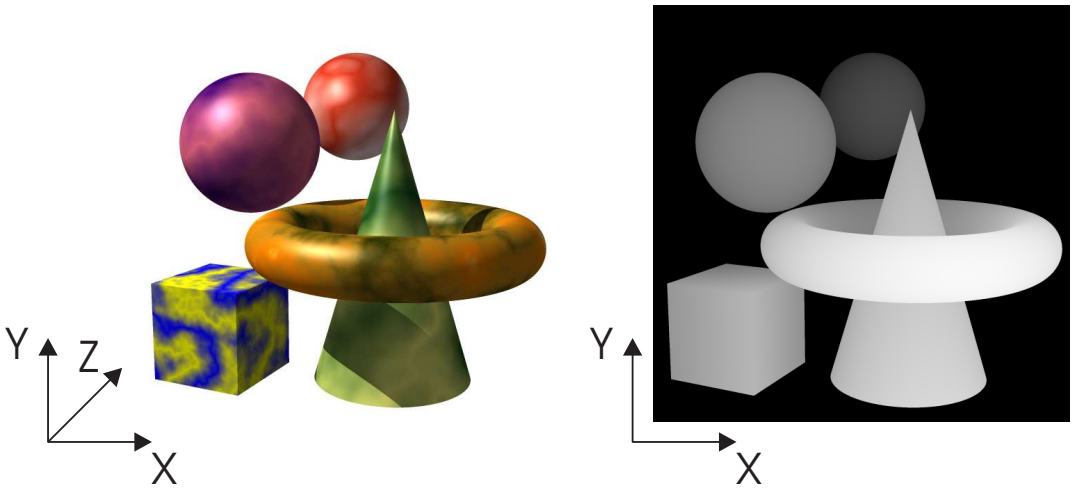


**Figure 2.2** (a) Volumetric model of the scene. Scene properties are represented as a function of 3D spatial position. (b) Depth-map scene model. The scene is modelled as a surface, whose depth is specified in terms of projected position onto a reference plane. In this example, as with the other depth-maps presented later in this thesis, depths are shown as inverse depths or disparities, with brighter points being closer to the camera and having a larger disparity.

than direct opacities.

Because of the large memory and computational requirements associated with representing and optimising a three dimensional volume, volumetric models have until recently seen little use in stereo reconstruction algorithms. This is changing however, and they are now becoming increasing popular especially in the rapidly developing field of computer graphics and animation [Seitz and Dyer 1999, Culbertson et al. 1999, De Bonet and Viola 1999, Kutulakos 2000, Eisert et al. 1999, Slabaugh et al. 2000b]. With such applications, a comprehensive radiometric model of the scene is required, that must be defined over a wide range of viewing directions. Because of their ability to represent general scenes independently of the cameras, volumetric models are now the most common choice for stereo reconstruction involving a large number of cameras from widely varying positions [Seitz et al. 2006].

Volumetric models can be grouped into either continuous or discrete models. With a continuous or piecewise continuous representation [Ilic and Fua 2006, Fua and Leclerc 1995, Carr et al. 2003, Faugeras and Keriven 1998], the desired radiometric properties are defined everywhere throughout the scene volume. In theory, this type of representation can have an infinite resolution. However, the scene must lie within the space defined by the chosen parametric functions. This space is a limited subset of all real scenes and so the resulting model will generally be an approximation to the real world.

In practice, many continuous scene models are parameterised using a finite set of control points which describe the scene properties. These control points may be arranged in a fixed grid, or allowed to vary their position so as to more closely represent the scene. With the fixed approach, the resulting model is equivalent to a discrete

representation with a defined interpolation function. The second approach, of using continuously movable control points, is more complex but has the advantage that a more accurate scene estimate can be obtained with the same number of parameters.

With a continuous representation the first and second order derivatives are generally defined and can usually be calculated relatively easily, lending itself to efficient optimisation through partial differential equations. While this technique is useful, the obtained solution is usually only locally optimal. The biggest problem with using a continuous representation is that it is usually more difficult to represent and optimise. For this reason most stereo algorithms, including the work presented in this thesis, use a discrete approach.

In a discrete representation, properties of the scene are represented at a finite set of sample points distributed throughout the scene volume. These sample points are usually referred to as voxels in most stereo literature. To prevent aliasing and help relate the scene model to the camera data, bandlimited values are represented. Associated with each voxel is a convolution kernel, which defines the bandwidth or smoothing of the sampled radiometric properties at that point in the scene. Given a set of voxels, the radiometric value at any intermediate position can be obtained or estimated through interpolation.

Voxels can be distributed in a variety of ways. The simplest of these is uniform sampling, where voxels are evenly distributed on a regular grid throughout the scene volume. This approach is commonly used for reconstructing individual objects, or objects contained within a small finite volume where a constant resolution is desired. The other common sampling scheme is disparity sampling, where voxels are located on planes spaced at constant intervals in inverse depth relative to some reference plane. This sampling method is usually adopted when the cameras are located near to one another and face in approximately the same direction. An advantage of this approach is that semi-infinite scenes can easily be modelled, using a small finite number of parameters. It also has the property that the scene resolution is proportional to its projected area onto the reference plane. The resolution of nearby, and so visually more significant, objects is therefore higher. This corresponds closely to human visual perception. In addition to these two standard approaches, a variety of other sampling distributions can be used. This can be useful in situations when the cameras are positioned arbitrarily throughout the scene, as it allows the resolution within regions of interest or close to each camera to be higher.

### 2.1.2 Depth-map models

In contrast to volumetric models, depth-map models represent the scene as a surface whose depth relative to some plane in space is a function of its projected position onto that plane (Fig. 2.2(b)). These depths are often described or presented in terms of

inverse depths or disparities, as these correspond more closely with the resolvable resolution limits of the scene. The depth-map approach is particularly suited to applications such as aerial photogrammetry, where the objective is to determine a height or depth map of the terrain surface. It is also useful in many robotic applications where the depth of visible objects needs to be ascertained.

With traditional two camera stereo algorithms the depth-map is usually defined relative to one or other of the camera images. In this case, scene points are mapped onto the reference plane through perspective projection. The depth-map representation is easily extended to multiple cameras, by simply choosing one of the images as the reference image.

Instead of defining the depth-map relative to a physical camera or image, it can also be defined relative to a virtual camera. This virtual camera can have an arbitrary position and projection function. The use of a virtual camera is common in many real stereo imaging systems. A typical example is the generation of digital elevation maps from aerial photographs, where it is desirable to generate a depth-map relative to a vertical orthographic projection, or ortho-image.

As with volumetric models, depth-map models can be either discrete or continuous. With a discrete model, scene depths are represented using a finite number of points spread over the reference plane. This is the most common approach since it provides a simple, easy to use, representation of the scene. With a discrete model, the depth at each sample point corresponds to the average or bandlimited depth over some windowed region in the image plane. Usually this depth is represented discretely, although continuous values are sometimes used. Another common approach is to model the depth-map using a continuous or piecewise continuous surface. The main advantage of this approach is that surface normals can be calculated easily and accurately, allowing reflectance and smoothness priors to be more easily incorporated. However, such representations are more complex and difficult to optimise, especially when there are discontinuities within the scene. Continuous optimisation techniques are also prone to getting stuck in local minima. For these reasons continuous surface models are usually more suited to reconstructing smooth continuous surfaces or those scenes where a good initial estimate is already known.

The depth-map scene model offers a number of advantages over a more comprehensive volumetric model. Firstly, it is a more compact representation and therefore uses less memory. This helps reduce the computation time. It is also often easier to optimise the resulting probability function as it usually has a simpler structure. The depth-map model also enforces directional ‘completeness’, whereby every pixel in the reference image is assigned some depth. This is a useful property, since the resulting reconstruction will be complete when viewed from the reference position. Completeness can be achieved with a volumetric model, but this requires additional complex priors involving multiple variables, thereby complicating the optimisation process.

Unfortunately there are also problems and limitations when using a depth-map model of the scene. These are usually associated with the fact that there is an inherent loss of information when representing a three-dimensional scene as a two-dimensional surface. For example, the interior of objects cannot be represented, nor can any surface occluded from view in the reference image. In many cases this is not much of a problem, since interior or fully occluded points do not affect the observed image data. However, problems do arise with semi-occluded points which are visible in one or more images but not in the reference image. Semi-transparent regions also cause problems, where interior points are visible in one or more of the camera images. Although fully occluded points do not affect the image data, they can affect the overall probability of the scene estimate depending on the prior knowledge that is applied. In many instances, information regarding spatial smoothness or continuity involves all points, not just those that are visible. With a depth-map model, such prior information cannot be applied directly and must therefore be approximated by some function over the depth-map. Again, this leads to a reduction in information and consequently the quality and reliability of the scene estimate is decreased.

In addition to representation limitations, the depth-map model complicates the mapping between scene and image parameters. Any discontinuities in depth across the reference image will appear as undefined regions when viewed from an oblique or perpendicular angle. This is a problem when trying to recreate views of the scene from any position other than that of the reference image, as some pixels will not correspond with any scene points. It is also a problem when it comes to making full use of the available image data, since some image pixels may be independent of the estimated surface and so cannot contribute to the reconstruction process.

To overcome some of these limitations a multiple depth-map model can be used. Here the scene is represented as a collection of depth-maps relative to the various camera images. This approach is used by a number of researchers [Szeliski and Golland 1999, Sun et al. 2005] and allows all the image data to be used equally. The resulting scene estimate is complete with respect to all the images but may still be incomplete from other positions. This greatly improves the reconstruction process since full use is made of the image data. Unfortunately, many of the other problems inherent to the single depth-map model still apply. In addition, extra constraints must be applied so that various depth-maps are consistent with one another.

## 2.2 IMAGE FORMATION

Given a particular scene model, the next step is to relate the set of scene parameters to the observed sensor data. This relationship is governed by the process of image formation and is usually expressed in terms of a mapping function from scene parameters to image parameters. With photometric images the recorded pixel values give

a measure of the incident intensity at different points on the imaging surface. These intensities, in turn, are related to the scene radiances and transmittances through the optical properties of the camera.

The process of image formation can be considered in several parts: geometric mapping, radiometry, and sensor measurement. The first of these, geometric mapping, describes the projection of three dimensional scene coordinates onto the image plane. The system radiometry then describes the relationship between scene radiances or intensities and the projected image intensities. Finally, pixel values are related to the incident image intensities through the process of sensor measurement.

The resulting mapping involves a collection of triple integrals over various regions in scene space. This is rather complex and extremely difficult to invert. In addition, for any given scene parametrisation the radiances and opacities will either be undefined at most points, or simply an approximation to the real world. Therefore, the value at arbitrary points must be estimated from the given parameters, usually through some form of interpolation. This introduces uncertainty and additional complexity into the reconstruction process. Consequently, the image formation process is nearly always simplified to help improve the reconstruction process.

### 2.2.1 Geometric mapping

The function of the camera optics is to form a projection of the scene on the imaging surface. The projected intensities are then converted into a recorded image by the sensor elements. Referred to as forward projection, or simply projection, this process can be described as a mapping from three dimensional scene coordinates to two dimensional image coordinates. Because of the reduced dimensionality, there is a fundamental loss of information. This results in a many to one mapping, where an entire line or ray in scene space is projected onto the same point in image space. The projection process is consequently irreversible, as a unique inverse does not exist.

The simplest possible camera consists of a pinhole and imaging plane (Fig. 2.3). A theoretical pinhole only allows through a single ray of light from each scene point. Consequently, every scene point will be in clear focus, since there is no requirement to focus multiple rays of light into a single point on the image plane. Assuming geometric optics [Born and Wolf 1980], a point $(X, Y, Z)$ in scene space will map or project to a point $(x, y)$ in image space. This mapping is described by the perspective transform,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \frac{d_i X}{Z} \\ \frac{d_i Y}{Z} \end{bmatrix}, \tag{2.1}$$

where $d_i$ is the distance between the image and the principal plane.

With multiple cameras, a common set of coordinates that are independent of camera position is required. This is necessary to describe the position and orientation of each

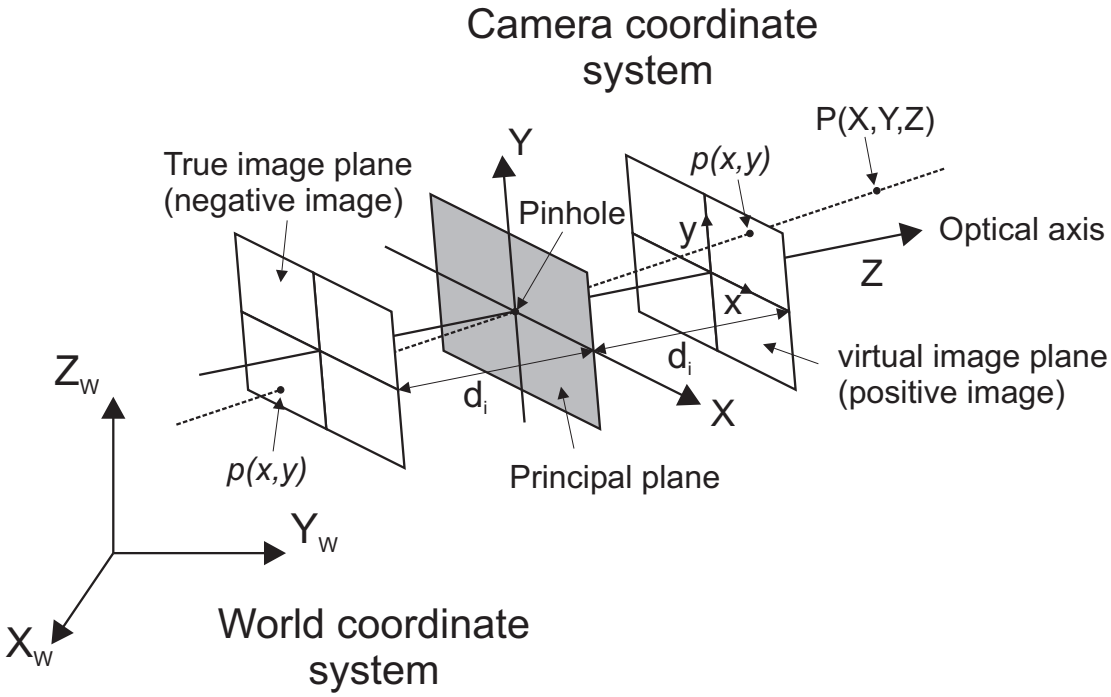


**Figure 2.3** Ideal pinhole camera model, showing the projected position $p(x, y)$ of a scene point $P(X, Y, Z)$ on the image plane. The projected image will appear inverted with respect to the scene coordinates, and must therefore be inverted to appear correctly. This is equivalent to projecting the scene onto a virtual plane in front of the camera. With multiple cameras, a common set of world coordinates $(X_W, Y_W, Z_W)$ must be used to relate spatial positions between the cameras.

camera as well as to relate 3D points between cameras. Referred to as world coordinates, see Fig. 2.3, these are related to camera coordinates by the geometric transform,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \boldsymbol{T}, \tag{2.2}$$

where $\boldsymbol{T}$ is a translation vector,

$$\boldsymbol{T} = \Big[T_X, T_Y, T_Z\Big]^T, \tag{2.3}$$

and $\boldsymbol{R}$ is a rotation matrix,

$$\boldsymbol{R} = \boldsymbol{R}_X \boldsymbol{R}_Y \boldsymbol{R}_Z = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}. \tag{2.4}$$

The rotation matrix $\boldsymbol{R}$ is the result of three partial transforms $\boldsymbol{R}_X$, $\boldsymbol{R}_Y$, and $\boldsymbol{R}_Z$, which describe the rotation about the $X_w$, $Y_w$ and $Z_w$ axes respectively. It is important to note that the order in which these partial transforms are applied is crucial. The matrix

$\boldsymbol{R}_X$ is a function of the angle of rotation $\theta_X$ about the $X_w$ axis and is given by

$$\boldsymbol{R}_X(\theta_X) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_X) & \sin(\theta_X) \\ 0 & -\sin(\theta_X) & \cos(\theta_X) \end{bmatrix}. \tag{2.5}$$

The matrix $\boldsymbol{R}_Y$ is a function of the angle of rotation $\theta_Y$ about the $Y_w$ axis, and is given by

$$\boldsymbol{R}_Y(\theta_Y) = \begin{bmatrix} \cos(\theta_Y) & 0 & -\sin(\theta_Y) \\ 0 & 1 & 0 \\ \sin(\theta_Y) & 0 & \cos(\theta_Y) \end{bmatrix}. \tag{2.6}$$

The matrix $\boldsymbol{R}_Z$ is a function of the angle of rotation $\theta_Z$ about the $Z_w$ axis, and is given by

$$\boldsymbol{R}_Z(\theta_Z) = \begin{bmatrix} \cos(\theta_Z) & \sin(\theta_Z) & 0 \\ -\sin(\theta_Z) & \cos(\theta_Z) & 0 \\ 1 & 0 & 0 \end{bmatrix}. \tag{2.7}$$

So far an ideal pinhole camera model has been assumed, however, in reality a pinhole camera must have a finite sized hole or aperture if a usable amount of light is to be let through. In this case, a cone of rays is observed from any point in scene space rather than a single ray. Points will therefore be imaged as a small circle instead of a point (see Fig. 2.4(a)). To reduce the size of this circle, referred to as a blur circle, the aperture diameter must be reduced. However, this will decrease the amount of light let through. In addition, because of diffraction effects, the size of the aperture can only be reduced to the order of a few wavelengths before the blurring caused by fringing becomes more significant than the blurring caused by diverging rays. At this point, any further reduction in the aperture diameter will actually increase the effective blur diameter and the imaging system is said to be diffraction limited. To overcome these problems, an optical lens is used instead of just a simple aperture. The lens focuses the divergent incoming rays to a single point some distance behind the principal plane. Assuming a thin lens approximation [Born and Wolf 1980], this distance is given by the Gaussian lens equation,

$$\frac{1}{d_i} = \frac{1}{f} - \frac{1}{Z}, \tag{2.8}$$

where $f$ is the focal length of the lens, $d_i$ is the distance of the image behind the principal plane, and $Z$ is the distance of the object in front of the principal plane. If the image plane is positioned at this depth, then a point in scene space will project to a single point in image space (see Fig. 2.4(b)). The image coordinates of this point are the same as those obtained through an ideal pinhole camera, and are given by the perspective transform Eq. 2.1.

The problem with using a lens is that for a given image plane position, scene points will only be in sharp focus along some corresponding conjugate plane in scene space.

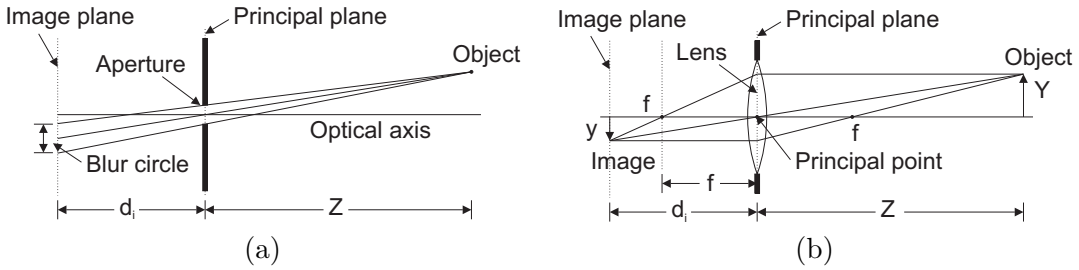


**Figure 2.4** (a) With a real pinhole camera, radiating object points will project to a circle on the image plane. This results in a blurred image, and is equivalent to convolving the ideal projected image intensities with a circular blurring function. (b) Thin lens approximation. By using a convex lens, the diverging rays from a given scene point can be focused into a single point on the image plane. This produces a sharp image of all points located a set distance from the camera, as given in Eq. 2.8.

As the deviation from this plane increases, scene points will become progressively more blurred in image space. For an acceptable blur circle diameter, there is a corresponding range of depths in scene space within which points must lie. This range of depths, called depth-of-view, is given by

$$\text{DOV} = \frac{Bf^2 d_i R}{(d_i - f)^2 R^2 - f^2 B^2/4}, \tag{2.9}$$

where the depth-of-view (DOV) is a function of aperture radius $R$, focal length $f$, image plane position $d_i$, and acceptable blur diameter $B$. In the case of digital cameras, image resolution is already limited by the spacing within the photo-sensor array and so an acceptable blur circle diameter is usually defined as being equal to the sensor spacing.

The other problem with using a lens is that it will introduce a number of distortions that affect the projection and radiometry of the camera. Luckily these are usually minor if good quality optics are used, and the resulting errors can be largely corrected by applying an appropriate image transform prior to stereo matching.

Although perspective projection is used by the large majority of cameras, other types of projection are possible. For instance, instead of projecting onto a flat plane, it is also possible to project onto a curved surface. This occurs in most biological imaging systems, such as that of the human eye. A big advantage of this approach is that the field of vision is increased for the same imaging area. The same effect can be achieved with a flat sensor by using a special fisheye lens. Alternatively the scene can be imaged through a curved mirror, or rotationally scanned. This approach is often used in applications where wide panoramic views are required.

### 2.2.2 Radiometry

Having established a geometric mapping from scene coordinates to image coordinates, the light intensity on the image plane can be determined from the scene radiances. This relationship is derived for an ideal camera lens that is perfectly in focus, assuming a

standard perspective projection. To simplify the analysis, projective coordinates will be used for describing the position of scene points relative to the camera. With this coordinate system a pointś position $(x, y, Z)$ is described by its depth $Z$ as well as its projected position $(x, y)$ in the image plane. Projective coordinates $(x, y, Z)$ can easily be transformed to standard Cartesian scene coordinates $(X, Y, Z)$ using the perspective transform given in Eq. 2.1.

First, consider the intensity contribution on the image plane from an infinitesimal volume $\delta V$ in scene space, see Fig. 2.5(a). The light intensity or *irradiance* $\delta I$ falling on the image plane due to $\delta V$, measured in power per unit area ($\mathrm{Wm}^{-2}$), is found by dividing the total power or *radiant flux* $\delta W$ received from $\delta V$ by the area $\delta S$ over which it is spread. For an approximate point source the radiant flux is related to the volumetric intensity of the source by

$$\delta W = DT\Omega\delta V, \tag{2.10}$$

where $D$ is the volumetric intensity measured in watts per steradian per unit volume ($\mathrm{W\,srad}^{-1}\mathrm{m}^{-3}$), $T$ is the transmittance from the source to the camera, and $\Omega$ is the solid angle extended by the camera lens as viewed by the source. $D$ and $T$ are functions of both the source and camera position and so will vary throughout the scene and between camera images.

The transmittance $T$ is defined as the ratio of transmitted intensity at the camera to the source intensity, for parallel rays travelling between the source and the camera. This varies between 0 and 1, depending on the opacity of the intervening medium. In stereo literature this is commonly referred to as the visibility of the scene point. The transmittance can be described in terms of per unit transmittances throughout the scene. Dividing the path from the source to the camera into a number of sections, the total transmittance is found by multiplying the transmittances of each subsection together. Taking logarithms, this gives a summation along the length of the path. A general expression for $T$ can then be obtained by taking the limit as the section length $\delta r \to 0$, giving

$$\log(T(x, y, Z)) = \int_{r=0}^{Z/\cos\theta} \log(\tau(x, y, Z))\,\mathrm{d}r, \tag{2.11}$$

where $tau$ is the per unit transmittance within scene space, $\theta$ is the angle between the source and the principle axis, and $r$ is the distance to the camera. By integrating with respect to $Z$, and using the dummy variable of integration $w = r\cos\theta$, this can equivalently be written as

$$\log(T(x, y, Z)) = \frac{1}{\cos\theta}\int_{w=0}^{Z} \log(\tau(x, y, w))\,\mathrm{d}w \tag{2.12}$$

This is commonly expressed in terms of optical density $o = -\log(\tau)$.

The solid angle $\Omega$ is defined to be the surface area of the projection of the lens onto

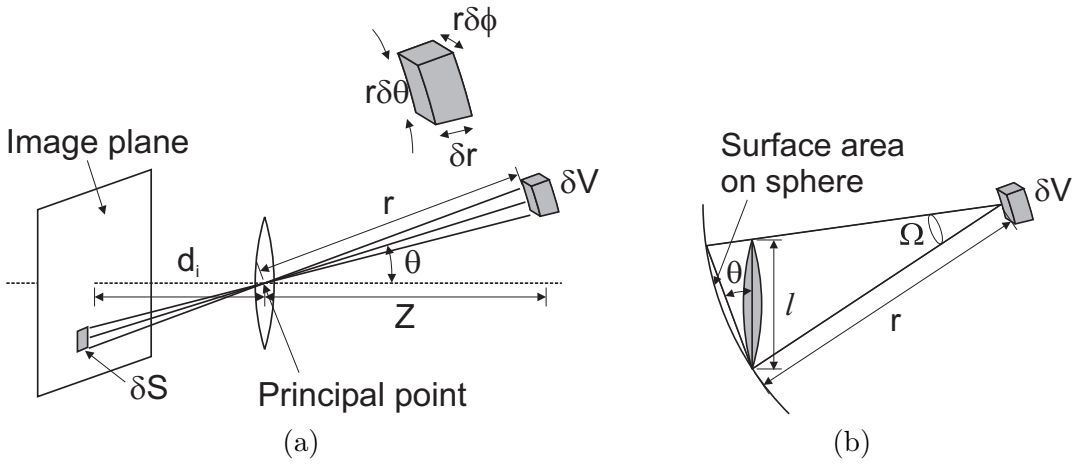


**Figure 2.5** (a) The light intensity $\delta I$ falling on the image plane due to radiating source $\delta V$ is equal to the total power $\delta W$ received from $\delta V$ divided by the area $\delta S$ over which it is spread. Using spherical coordinates, $\delta V$ can be expressed as $r\delta\theta r\delta\phi\delta r$. (b) The solid angle $\Omega$, subtended by the lens at the point source, is defined as the projected area of the lens onto a sphere centred at that point, divided by the square of the sphere's radius, $r$. For small angles this is approximately equal to $A\cos\theta$, where $A$ is the area of the lens viewed front on. For a circular lens $A = \pi l^2/4$, where $l$ is the aperture diameter.

a unit sphere centred at the point source. This can be calculated by projecting onto a sphere of radius $r$, then dividing the projected area by $r^2$, see Fig. 2.5(b). If $\Omega$ is relatively small, the projected area will be approximately equal to $A\cos\theta$, where A is the area of the lens viewed front on. Assuming a circular lens with aperture diameter $l$, the solid angle subtended by the lens can be expressed as

$$\Omega = \frac{\pi}{4}\frac{l^2\cos\theta}{r^2}. \tag{2.13}$$

Substituting this into Eq. 2.10 and using spherical coordinates to replace $\delta V$ with $r\delta\theta r\delta\phi\delta r$, the total flux intercepted by the lens and incident on the image plane is given by

$$\delta W = \frac{\pi}{4} l^2 \cos\theta DT\delta\theta\delta\phi\delta r. \tag{2.14}$$

To find the intensity on the image plane, the area $\delta S$ over which the radiant flux is spread must also be determined. Using simple geometry and assuming that the solid angle subtended by $\delta S$ from the principal point, $\delta\psi$, is sufficiently small, this can be expressed as

$$\delta S = \frac{\delta\psi}{\cos\theta}\left(\frac{d_i}{\cos\theta}\right)^2, \tag{2.15}$$

Since the rays passing through the principal point are not deflected, the solid angle of the cone leading to $\delta S$ is equal to the solid angle of the cone leading to $\delta V$. By equating solid angles we get $\delta\psi = \delta\theta\delta\phi$, allowing the area $\delta S$ to be expressed as

$$\delta S = \frac{d_i^2\delta\theta\delta\phi}{\cos^3\theta}. \tag{2.16}$$

The image irradiance can then be found by dividing $\delta W$ by $\delta S$, giving

$$\delta I = \frac{\pi}{4}\left(\frac{l}{d_i}\right)^2 \cos^4\theta D\delta r T. \tag{2.17}$$

This states that the intensity in the image plane is independent of the apparent area of $\delta V$ and drops off as $\cos^4\theta$. This nonlinear reduction in intensity with $\theta$ is called optical vignetting and results in a darkened circular border around an image. The effects of this can easily be corrected by appropriate scaling of the pixel values. In a digital camera this is often performed as part of the internal processing of an image.

To determine the overall irradiance at any point on the $i^{\text{th}}$ image plane, the irradiances due to each contributing infinitesimal volume need to be summed together. This requires that the light emitted from each point is non-coherent. This is true for all ordinary light sources, since the light comes from independently emitting atoms. However, in some special cases, such as laser light, the emitted photons are "in step" and have a definite phase relation. Using projective camera coordinates $(x, y, Z)$, and noting that $\theta$ is a function of $x$ and $y$, given by $\theta = \arctan(\sqrt{x^2+y^2}/d_i)$, this summation is equivalent to integrating over $Z$ and can be expressed as

$$\acute{I}_i(x,y) = \frac{\pi}{4}\left(\frac{l}{d_i}\right)^2 \cos^4(\theta)\int_{Z=0}^{\infty} D_i(x,y,Z)T_i(x,y,Z)\frac{1}{\cos(\theta)}\,\mathrm{d}Z, \tag{2.18}$$

where $\acute{I}_i(x,y)$ are the ideal incident intensities, $D_i(x,y,Z)$ is the volumetric intensity and $T_i(x,y,Z)$ are the transmittances as viewed from the $i^{\text{th}}$ camera. By appropriate scaling of the pixel values prior to reconstruction this can be simplified to give

$$\acute{I}_i(x,y) = \int_{Z=0}^{\infty} D_i(x,y,Z)T_i(x,y,Z)\,\mathrm{d}Z. \tag{2.19}$$

This gives the ideal mapping from volumetric intensities within a scene to surface irradiances or intensities on the image plane. Unfortunately this mapping will be inexact when applied to real scenes and cameras. Firstly, not all the points within the scene can be in exact focus. For a given camera setup, only those points lying on the focal plane will be focused accurately. Secondly, the camera lens will introduce a number of distortions or abberations that will further alter the incident intensities. Both of these factors will cause the mapping from volumetric intensities to image intensities to differ slightly from the ideal mapping. Such distortions can be modelled as a convolution of the ideal incident intensities with a suitable blurring function, plus the addition of image noise. The actual incident intensities can therefore be expressed as

$$I_i(x,y) = \acute{I}_i(x,y) \otimes b_i(x,y,u,v) + n_i(x,y), \tag{2.20}$$

where $b_i(x,y,u,v)$ is the spatially variant blurring function, $\otimes$ is the convolution operator, and $n_i(x,y)$ are additive modelling errors.

### 2.2.3 Image sensors

Having formed a relationship between the volumetric intensities in the scene and the incident intensities on the image plane, the next step is to relate these to the intensities recorded or measured by the camera. With a typical digital camera these measurements are made using an array of sensor elements, each generating a value relating to the total light flux falling on its surface. These sensor elements define a grid of regions on the image plane, represented as pixels in the recorded image. With colour cameras each region contains several sensor elements that measure the light flux within various spectral or colour bands. Usually three colour bands are used, as this corresponds with the human visual system. However, in some cameras four colour bands are used as this gives a greater range of colours. Assuming appropriate calibration of pixel intensities, the value of an individual pixel is found by averaging the incident intensity over the surface of the corresponding sensor element.

Using $x_s$ and $y_s$ to denote the $x$ and $y$ dimensions of each sensor element, the average intensity $\overline{I}_i$ measured by a sensor element centred at $x$ and $y$ on the image plane is

$$\overline{I}_i(x,y) = \frac{1}{x_s y_s} \int_{x-\frac{x_s}{2}}^{x+\frac{x_s}{2}} \int_{y-\frac{y_s}{2}}^{y+\frac{y_s}{2}} I_i(u,v)\,\mathrm{d}u\,\mathrm{d}v. \tag{2.21}$$

This can alternatively be expressed as

$$\overline{I}_i(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_i(u,v) \sqcap_s (x-u, y-v)\,\mathrm{d}u\,\mathrm{d}v, \tag{2.22}$$

where $\sqcap_s(x,y)$ is a normalised rectangular window function of dimensions $x_s$, $y_s$ and height $\frac{1}{x_s y_s}$. This is equivalent to convolving the incident intensities with $\sqcap_s(x,y)$. By substituting the expression for $I_i$, given in Eq. 2.20, into Eq. 2.22, and using the associative and distributive properties of convolution, the average intensity can be expressed as

$$\overline{I}_i(x,y) = \acute{I}_i(x,y) \otimes h_i(x,y,u,v) + \overline{n}_i(x,y), \tag{2.23}$$

where $h_i(x,y,u,v) = b_i(x,y,u,v) \otimes \sqcap_s(x,y)$ is the overall pixel blurring function, and $\overline{n}_i(x,y) = n_i(x,y) \otimes \sqcap_s(x,y)$ is the average noise across each sensor element. This states that the intensity measured by each sensor element can be found by convolving the ideal incident intensities with the blurring function $h_i(x,y,u,v)$.

By substituting the expression for ideal intensities $\acute{I}_i(x,y)$, given in Eq. 2.19, into Eq. 2.23 and changing the order of integration, the average intensity across each sensor element can be directly related with the scene parameters $D$ and $T$, giving

$$\overline{I}_i(x,y) = \int_{Z=0}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} D_i(u,v,Z) T_i(u,v,Z) h_i(x,y,u,v)\,\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}Z + \overline{n}_i(x,y). \tag{2.24}$$

Approximating $h_i(x,y,u,v)$ with a finite extent window, the resulting mapping is equiv-

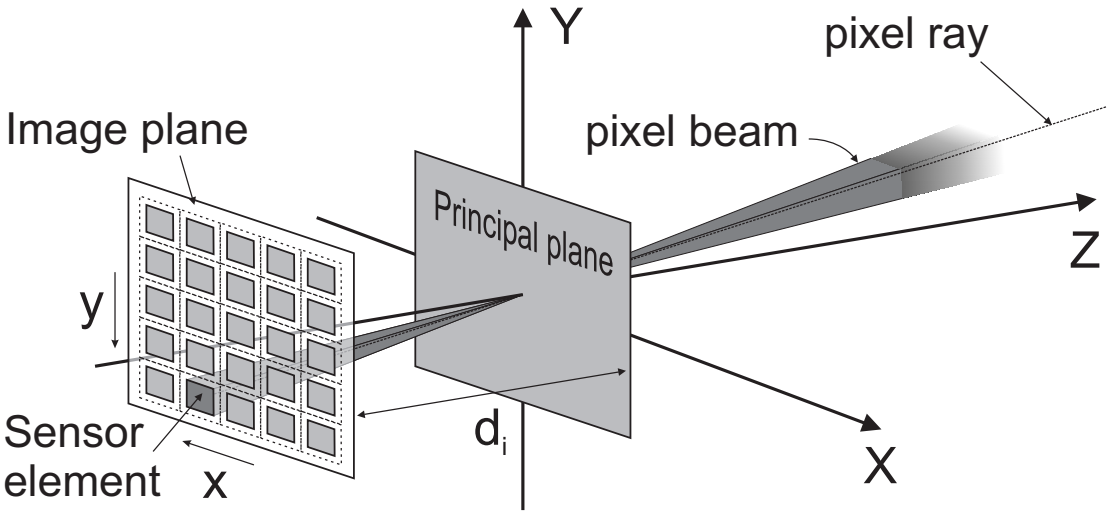


**Figure 2.6** The total light observed by a sensor element in the image plane is equal to the integral of visible scene radiances over a cone in scene space, referred to as the pixel beam. By appropriate smoothing of the scene intensities and transmittances, this integral can be approximated by a line integral along the centre of the pixel beam. This line, referred to as the pixel ray, is described by the mapping function given in Eq. 2.1.

alent to integrating over a cone in scene space, referred to as the pixel beam (see Fig. 2.6).

Having determined the average incident intensity, the sensor elements then convert the incident flux over each sensor into a recorded measurement. Although a number of different sensor technologies exist, they all follow the same basic operation. First, incoming photons are converted into a charge via the photo electric effect. The charge stored in each sensor element is then transferred out of the sensor array and converted into a voltage. This is finally converted into a digital measurement and processed to correct any known errors.

Throughout this process, noise and other errors are introduced into the intensity measurement. Referred to as image noise, these variations complicate the reconstruction process, as a degree of uncertainty is introduced into the mapping function. This introduces a number of problems as the incident intensities, and hence any related scene parameters, cannot be determined exactly from the image data. An important consequence of this is that the recorded intensities from a single point or surface will appear slightly different in each image. Therefore, if an accurate scene estimate is to be obtained, image noise must be accounted for and dealt with appropriately.

### 2.2.4 Image noise

The three main causes of image noise are photon noise, sensor noise, and quantisation noise [Healey and Kondepudy 1994]. The first of these, photon noise, arises from the

quantum nature of light and has a Poisson distribution,

$$\Pr(p|\rho,t) = \frac{(\rho t)^p e^{-\rho t}}{p!}, \tag{2.25}$$

where $p$ is the number of photons, $t$ is the observation time, and $\rho$ is the average intensity parameter measured in photons per second. Therefore, the expected intensity variation will depend upon the observed image intensities. For bright signals or long integration times, such as those encountered in standard daytime photography, the peak of this distribution is extremely sharp relative to the mean and noise fluctuations due to photon statistics can be ignored.

In addition to photon variation, noise will be introduced by the sensor itself. This can be decomposed into thermal noise, electronic noise, amplifier noise, and quantisation noise. Although these all have their own probability distributions, the resulting overall sensor noise can usually be closely approximated by a robust Gaussian distribution [Cortelazzo et al. 1994]. This is similar to a standard Gaussian or normal distribution, except that large deviations are given a higher probability. The resulting distribution can be described by its mean, variance and a robustness parameter.

Although the measured intensity variations are unknown and occur randomly, information about the likely distribution of the image noise can be obtained and is useful in solving the stereo problem. By formulating stereo reconstruction as a statistical optimisation problem, the noise probability distribution can be used to determine the likelihood of a given scene estimate. Such information about the distribution of the noise is usually obtained prior to scene reconstruction, either through experimentation or from sensor data sheets.

In addition to image noise, modelling errors or approximations introduce further discrepancies between the modelled incident intensities and the actual camera data. These variations can be treated as additional image noise, due to their pseudo-random nature. The resulting combined image noise $N_i(x,y)$ is usually modelled as a robust Gaussian. Camera pixel intensities $C_i(x,y)$ can therefore be expressed as the sum of the average incident intensity, given in Eq. 2.24, and the combined noise at each pixel, resulting in

**Theorem 1**

$$C_i(x,y) = \int_{Z=0}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D_i(u,v,Z)T_i(u,v,Z)h_i(x,y,u,v)\,\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}Z + N_i(x,y).$$

## 2.3 MAPPING SIMPLIFICATION

Given the mapping from scene intensities and transmittances to image data, the next step is to relate these radiometric properties to the scene parameters. This introduces a variety of problems depending on the scene model that is used. With a discrete model the value at any arbitrary scene point must be estimated or interpolated from the known sample points. This is impossible to do accurately with any real scene, as the bandwidth of the intensity and transmittance will always be much higher than the sampling rate. A continuous model on the other hand defines values throughout the scene. However, this will at best be an approximation to the real world, due to representational limitations of using a finite number of parameters.

Even if the scene intensities and transmittances could be obtained at every point, the resulting mapping would still involve the complex triple integral, given in Theorem 1. The inverse to this is highly ill-posed, due to the one-to-many inverse mapping from camera intensities to scene parameters. Therefore, regularisation or the use of additional information is required to constrain the solution. Even with such constraints, finding a near optimal inverse solution is exceeding difficult, as the relationship between scene points and image pixels is complex and involves a high level of interaction between the various model parameters.

This section presents a novel system model to simplify the mapping from scene parameters to image data, as well as providing a detailed derivation of this model. By describing the image formation process in terms of locally averaged or bandlimited values, rather than individual point values, it is shown that the triple integral in Theorem 1 can be approximated by a one dimensional line integral, or discrete summation, along the pixel rays.

### 2.3.1 Pixel ray integration

The triple integral in Theorem 1 can be approximated by a one dimensional line integral along the pixel rays by describing the camera pixel intensities in terms of low-pass filtered scene intensities and transmittances.

Assuming the volumetric intensity $D_i(u, v, Z)T_i(u, v, Z) = 0$ in the close vicinity of any camera, the integral in Theorem 1 will remain the same after convolving the integrand with a normalised depth invariant window function $\gamma_i(x, y, u, v, Z - w)$ in the $Z$ direction. This allows the measured camera pixel intensity to be equivalently expressed as

$$C_i(x, y) = \int_{Z=0}^{\infty} \xi_i(x, y, Z)\, \mathrm{d}Z + N_i(x, y), \tag{2.26}$$

where

$$\xi_i(x, y, Z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D_i(u, v, w)T_i(u, v, w)W_i(x, y, Z, u, v, w)\, \mathrm{d}u\, \mathrm{d}v\, \mathrm{d}w, \tag{2.27}$$

and

$$W_i(x, y, Z, u, v, w) = h_i(x, y, u, v)\gamma_i(x, y, u, v, Z - w). \tag{2.28}$$

Original proof of this is given in Appendix A. The term $\xi_i(x, y, Z)$, represents the low-pass filtered scene intensities that are transmitted to the $i^{\text{th}}$ camera. These are obtained by convolving the product of $D_i(x, y, Z)$ and $T_i(x, y, Z)$, with the spatially dependent imaging convolution kernel $W_i(x, y, Z, u, v, w)$. Using vector variables $s = (x, y, Z)$ and $t = (u, v, w)$, this can be expressed as

$$\xi_i(s) = \int_{-\infty}^{\infty} D_i(t)T_i(t)W_i(s, t)\,\mathrm{d}t. \tag{2.29}$$

By introducing the binary radiance operator, $K_i(t)$, where $K_i(t) = 0$ if $D_i(t) = 0$, and $K_i(t) = 1$ if $D_i(t) > 0$, the expression for $\xi_i(x, y, Z)$ can equivalently be written as

$$\xi_i(s) = \int_{-\infty}^{\infty} D_i(t)T_i(t)W_i(s, t)K_i(t)\,\mathrm{d}t. \tag{2.30}$$

Next, the scene transmittances and volumetric intensities can be re-expressed as the sum of a low-pass filtered value plus a difference term, giving

$$T_i(t) = \tilde{T}_i(s) + \acute{T}_i(s, t), \tag{2.31}$$

$$D_i(t) = \tilde{D}_i(s) + \acute{D}_i(s, t), \tag{2.32}$$

where

$$\tilde{T}_i(s) = \frac{1}{\int_{-\infty}^{\infty} W_i(s, t)K_i(t)\,\mathrm{d}t} \int_{-\infty}^{\infty} T_i(t)W_i(s, t)K_i(t)\,\mathrm{d}t, \tag{2.33}$$

$$\tilde{D}_i(s) = \frac{1}{\int_{-\infty}^{\infty} W_i(s, t)K_i(t)\,\mathrm{d}t} \int_{-\infty}^{\infty} D_i(t)W_i(s, t)K_i(t)\,\mathrm{d}t. \tag{2.34}$$

Substituting these into Eq. 2.30, and taking any terms that are independent of $t$ outside the integral, the term $\xi_i(s)$ can be written

$$\begin{aligned}
\xi_i(s) &= \int_{-\infty}^{\infty} (\tilde{D}_i(s) + \acute{D}_i(s, t))(\tilde{T}_i(s) + \acute{T}_i(s, t))W_i(s, t)K_i(t)\,\mathrm{d}t \\
&= \tilde{D}_i(s)\tilde{T}_i(s) \int_{-\infty}^{\infty} W_i(s, t)K_i(t)\,\mathrm{d}t \\
&\quad + \tilde{D}_i(s) \int_{-\infty}^{\infty} \acute{T}_i(s, t)W_i(s, t)K_i(t)\,\mathrm{d}t \\
&\quad + \tilde{T}_i(s) \int_{-\infty}^{\infty} \acute{D}_i(s, t)W_i(s, t)K_i(t)\,\mathrm{d}t \\
&\quad + \int_{-\infty}^{\infty} \acute{D}_i(s, t)\acute{T}_i(s, t)W_i(s, t)K_i(t)\,\mathrm{d}t.
\end{aligned} \tag{2.35}$$

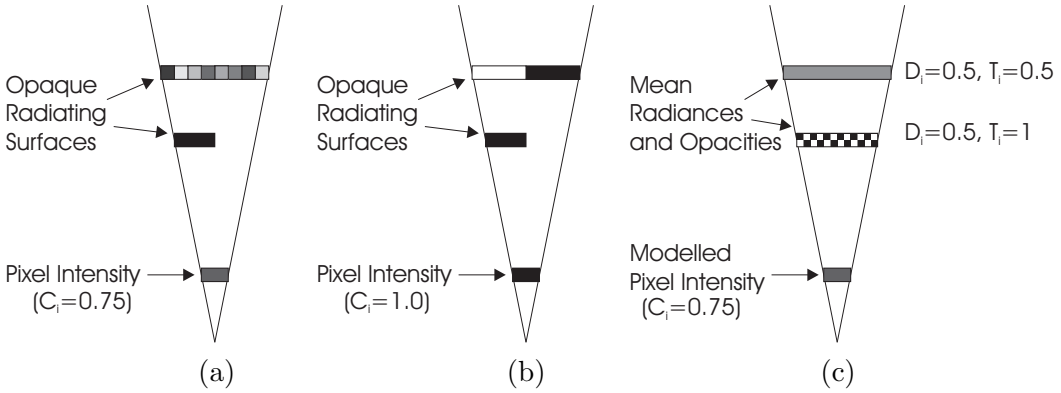


**Figure 2.7** In most situations the scene can be accurately modelled using low-pass filtered transmittances, $\tilde{T}_i$, and volumetric intensities, $\overline{D}_i$. (a) Assuming variations in $T_i$ and $D_i$ over the imaging kernel are uncorrelated, the low-pass filtered approximation, (c), will correspond closely with the image data. (b) In some instances, such as at transmittance boundaries, this assumption may be invalid, leading to errors in the predicted intensities (c).

This can be simplified by noting that

$$\begin{aligned}\int_{-\infty}^{\infty} \acute{D}_i(s,t)W_i(s,t)K_i(t)\,\mathrm{d}t &= \int_{-\infty}^{\infty} (D_i(t) - \tilde{D}_i(s))W_i(s,t)K_i(t)\,\mathrm{d}t \\ &= \int_{-\infty}^{\infty} D_i(t)W_i(s,t)K_i(t)\,\mathrm{d}t - \tilde{D}_i(s)\int_{-\infty}^{\infty} W_i(s,t)K_i(t)\,\mathrm{d}t \\ &= \tilde{D}_i(s)\int_{-\infty}^{\infty} W_i(s,t)K_i(t)\,\mathrm{d}t - \tilde{D}_i(s)\int_{-\infty}^{\infty} W_i(s,t)K_i(t)\,\mathrm{d}t \\ &= 0. \end{aligned} \tag{2.36}$$

And similarly,

$$\int_{-\infty}^{\infty} \acute{T}_i(s,t)W_i(s,t)K_i(t)\,\mathrm{d}t = 0. \tag{2.37}$$

Therefore, the second and third terms in Eq. 2.35 can be removed from the expression as they too will equal zero. The final term $\int_{-\infty}^{\infty} \acute{D}_i(s,t)\acute{T}_i(s,t)W_i(s,t)K_i(t)\,\mathrm{d}t$ will also equal zero, so long as $\acute{D}_i(s,t)$ and $\acute{T}_i(s,t)$ are uncorrelated under the convolution kernel $W_i(s,t)K_i(t)$. In most situations this will be approximately true, as either $\acute{D}_i(s,t)$ or $\acute{T}_i(s,t)$ will be small compared with the low-pass filtered terms, $\tilde{D}_i(s)$ and $\tilde{T}_i(s)$, or will approximate white noise (see Fig. 2.7).

The first term in Eq. 2.35 can also be simplified, giving

$$\begin{aligned}
&\tilde{D}_i(s)\tilde{T}_i(s)\int_{-\infty}^{\infty} W_i(s,t)K_i(t)\,\mathrm{d}t \\
&\quad= \frac{1}{\int_{-\infty}^{\infty} W_i(s,t)K_i(t)\,\mathrm{d}t}\int_{-\infty}^{\infty} D_i(t)W_i(s,t)K_i(t)\,\mathrm{d}t \times \tilde{T}_i(s)\int_{-\infty}^{\infty} W_i(s,t)K_i(t)\,\mathrm{d}t \\
&\quad= \tilde{T}_i(s)\int_{-\infty}^{\infty} D_i(t)W_i(s,t)K_i(t)\,\mathrm{d}t \\
&\quad= \tilde{T}_i(s)\int_{-\infty}^{\infty} D_i(t)W_i(s,t)\,\mathrm{d}t \\
&\quad= \tilde{T}_i(s)\overline{D}_i(s)\psi_{W_i}(s), \qquad (2.38)
\end{aligned}$$

where

$$\psi_{W_i}(s) = \int_{-\infty}^{\infty} W_i(s,t)\,\mathrm{d}t, \qquad (2.39)$$

and

$$\overline{D}_i(s) = \frac{1}{\psi_{W_i}(s)}\int_{-\infty}^{\infty} D_i(t)W_i(s,t)\,\mathrm{d}t. \qquad (2.40)$$

With these approximations the term $\xi_i(s)$ can be expressed as

$$\xi_i(s) = \overline{D}_i(s)\tilde{T}_i(s)\psi_{W_i}(s). \qquad (2.41)$$

This can be further simplified by noting that

$$\begin{aligned}
\psi_{W_i}(s) &= \psi_{W_i}(x,y,Z) \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} W_i(x,y,Z,u,v,w)\,\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}w \qquad (2.42) \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_i(x,y,u,v)\gamma_i(x,y,u,v,Z-w)\,\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}w \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_i(x,y,u,v)\,\mathrm{d}u\,\mathrm{d}v\int_{-\infty}^{\infty}\gamma_i(x,y,u,v,Z-w)\,\mathrm{d}w. \qquad (2.43)
\end{aligned}$$

The first term in this expression is an integral over the 2D image blurring function $h_i(x,y,u,v)$. In most cases this integral will be approximately unity over the range of image coordinates $x, y$. If not, it can easily be forced to equal unity by appropriately scaling pixel intensities prior to reconstruction. The second term will also equal unity, as $\gamma_i(x,y,u,v,Z-w)$ is a normalised spatially invariant windowing function in the $Z$ direction. Therefore, $\psi_{W_i}(s)$ will equal 1. Using this result, and substituting Eq. 2.41 back into Eq. 2.26, the recorded pixel intensities can finally be expressed as

**Theorem 2**

$$C_i(x,y) = \int_{Z=0}^{\infty} \overline{D}_i(x,y,Z)\tilde{T}_i(x,y,Z)\,\mathrm{d}Z + N_i(x,y).$$

Theorem 2 states that the measured intensity at each pixel is equal to an integral of the low-pass filtered transmittance and volumetric intensity along the corresponding pixel ray. The term $\overline{D}_i(x, y, Z)$ represents the low-pass filtered volumetric intensity in the direction of the $i^{\text{th}}$ camera, while $\tilde{T}_i(x, y, Z)$ is the low-pass filtered transmittance towards the $i^{\text{th}}$ camera of all radiating points. This reduces the three dimensional integral given in Theorem 1 to a simpler one dimensional integral.

### 2.3.2 Discrete summation

Instead of expressing the pixel intensities as an integral, they can alternatively be expressed as a summation of discrete terms. Starting from Theorem 1, and replacing $Z$ with the dummy variable of integration $w$, the triple integral can be broken down into the summation of $K$ sub-integrals, giving

$$C_i(x,y) = \sum_{k=1}^{K} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{w=d_{i(k-1)}(x,y,u,v)}^{d_{ik}(x,y,u,v)} D_i(u,v,w) T_i(u,v,w) h_i(x,y,u,v) \,\mathrm{d}w \,\mathrm{d}u \,\mathrm{d}v + N_i(x,y), \tag{2.44}$$

where $d_{ik}(x, y, u, v)$ are discrete depths ranging from $d_{i0}(x, y, u, v) = 0$ to $d_{iK}(x, y, u, v) = \infty$. Using $Z_{ik}(x, y) = (d_{ik}(x, y, x, y) + d_{i(k-1)}(x, y, x, y))/2$ to represent the $Z$ coordinates of each subregion, and $g_i(x, y, Z_{ik}(x, y), u, v, w)$ to represent a unity rectangular function from $w = d_{i(k-1)}(x, y, u, v)$ to $d_{ik}(x, y, u, v)$, this can alternatively be expressed as

$$C_i(x,y) = \sum_{k=1}^{K} \iiint_{-\infty}^{\infty} D_i(u,v,w) T_i(u,v,w) h_i(x,y,u,v) g_i(x,y,Z_{ik}(x,y),u,v,w) \,\mathrm{d}u \,\mathrm{d}v \,\mathrm{d}w + N_i(x,y)$$

Following a similar approach to the continuous case, this can be written as

$$C_i(x,y) = \sum_{k=1}^{K} \iiint_{-\infty}^{\infty} D_i(u,v,w) T_i(u,v,w) W_i(x,y,Z_{ik}(x,y),u,v,w) \,\mathrm{d}u \,\mathrm{d}v \,\mathrm{d}w + N_i(x,y), \tag{2.45}$$

where in this instance, $W_i(x, y, Z_{ik}(x, y), u, v, w) = h_i(x, y, u, v) g_i(x, y, Z_{ik}(x, y), u, v, w)$. Using Eq. 2.27, and the result given in Eq. 2.41, this can in turn be expressed as

$$\begin{aligned} C_i(x,y) &= \sum_{k=1}^{K} \xi_i(x,y,Z_{ik}(x,y)) + N_i(x,y) \\ &= \sum_{k=1}^{K} \overline{D}_i(x,y,Z_{ik}(x,y)) \tilde{T}_i(x,y,Z_{ik}(x,y)) \psi_{W_i}(x,y,Z_{ik}(x,y)) + N_i(x,y), \end{aligned} \tag{2.46}$$

where $\overline{D}_i(x, y, Z_{ik}(x, y))$ and $\tilde{T}_i(x, y, Z_{ik}(x, y))$ are the low-pass filtered intensities and transmittances defined by the convolution kernel $W_i(x, y, Z_{ik}(x, y), u, v, w)$.

### 2.3.3 Scene radiances

So far, the scene has been described in terms of volumetric intensities and transmittances. This gives a general model that is applicable over a wide spectrum of electromagnetic frequencies. However, in most situations the scene will consist of radiating surfaces, rather than diffuse radiating volumes. The volumetric intensity at such a surface is infinite and can be modelled in the vicinity of the surface using a weighted Dirac delta in the $Z$ direction, giving

$$D_i(x, y, Z) = R_i(x, y, Z) \sum_{n \in \xi_i(x,y)} \delta(Z - w_{in}(x, y)), \tag{2.47}$$

where $\xi_i(x, y)$ denotes the index of surfaces along the pixel ray $(x, y)$ of the $i^{\text{th}}$ camera, $R_i(x, y, Z)$ is the surface radiance in the direction of the $i^{\text{th}}$ camera, and $w_{in}(x, y)$ is the depth of the $n^{\text{th}}$ surface as a function of projected image position. This is clarified in Fig. 2.8. Surface radiance is defined as radiant intensity per unit projected area in a radial direction. Substituting this into Eq. 2.40, the low-pass filtered intensity $\overline{D}_i(x, y, Z)$, can be expressed as

$$\begin{aligned}
\overline{D}_i(x, y, Z) &= \frac{1}{\psi_{W_i}(x, y, Z)} \times \\
&\quad \iiint_{-\infty}^{\infty} \left( R_i(u, v, w) \sum_{n \in \xi_i(x,y)} \delta(w - w_{in}(u, v)) \right) W_i(x, y, Z, u, v, w) \, \mathrm{d}w \, \mathrm{d}u \, \mathrm{d}v \\
&= \frac{1}{\psi_{W_i}(x, y, Z)} \iint_{-\infty}^{\infty} \sum_{n \in \xi_i(x,y)} R_i(u, v, w_{in}(u, v)) W_i(x, y, Z, u, v, w_{in}(u, v)) \, \mathrm{d}u \, \mathrm{d}v \\
&= \frac{1}{\psi_{W_i}(x, y, Z)} \overline{R}_i(x, y, Z),
\end{aligned} \tag{2.48}$$

where $\overline{R}_i(x, y, Z)$ is the low-pass filtered surface radiance, obtained by performing a weighted surface integral over the radiating surfaces. Finally, substituting this into Eq. 2.46, the pixel intensities can be written as

$$C_i(x, y) = \sum_{k=1}^{K} \overline{R}_i(x, y, Z_{ik}(x, y)) \tilde{T}_i(x, y, Z_{ik}(x, y)) + N_i(x, y). \tag{2.49}$$

This defines the measured pixel intensities as a discrete summation of the low-pass filtered radiances and transmittances.

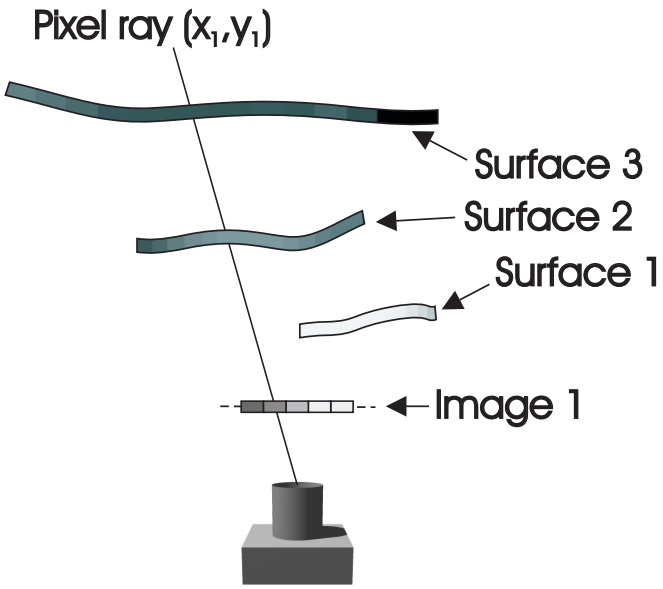


**Figure 2.8** A scene consisting of three radiating surfaces. In this example there are two surfaces, surface 2 and surface 3, which lie along the pixel ray $(x_1, y_1)$. Therefore, using Eq. 2.47 the volumetric intensity along this ray is given by $D_1(x_1, y_1, Z) = R_1(x_1, y_1, Z)\,(\delta(Z - w_{12}(x_1, y_1)) + \delta(Z - w_{13}(x_1, y_1))$, where $w_{1i}(x_1, y_1)$ is the depth of surface $i$ along pixel ray $(x_1, y_1)$.

### 2.3.4 Imaging convolution kernel

Although Eq. 2.49 considerably simplifies the mapping, it introduces two main problems. The first complication is that the set of discrete scene points that correspond with the pixels in one image will not usually correspond with pixels in another image. This is particularly the case when more than two cameras are used, since it is usually impossible to select a common set of points that project to the exact pixel locations in every image. Consequently, some form of interpolation is needed to map between the discrete mapping points for each image and a common set of sample points. This increases the complexity of the mapping, as pixel intensities will depend on numerous scene points, not just those lying along a ray in space.

The second, and more significant problem, is that the weighted region of integration, defined by the imaging convolution kernel, $W_i(x, y, Z, u, v, w)$, is a function of camera position and optics. Therefore, the values $\overline{R}_i(x, y, Z)$ and $\tilde{T}_i(x, y, Z)$ will vary between images due to differences in the convolution kernel. This further complicates the system model and makes it difficult to compare the data between different cameras.

To transform from a common set of samples with a defined kernel, to the set of imaging samples for each camera, the underlying radiometric function, represented by the samples, must be filtered so that the resulting function is equivalent to convolving the scene with the imaging kernel for each camera. Because of a general loss of information in the convolution process, this cannot be achieved exactly, especially in the presence of image noise. The other difficulty with filtering, other than adding ad-

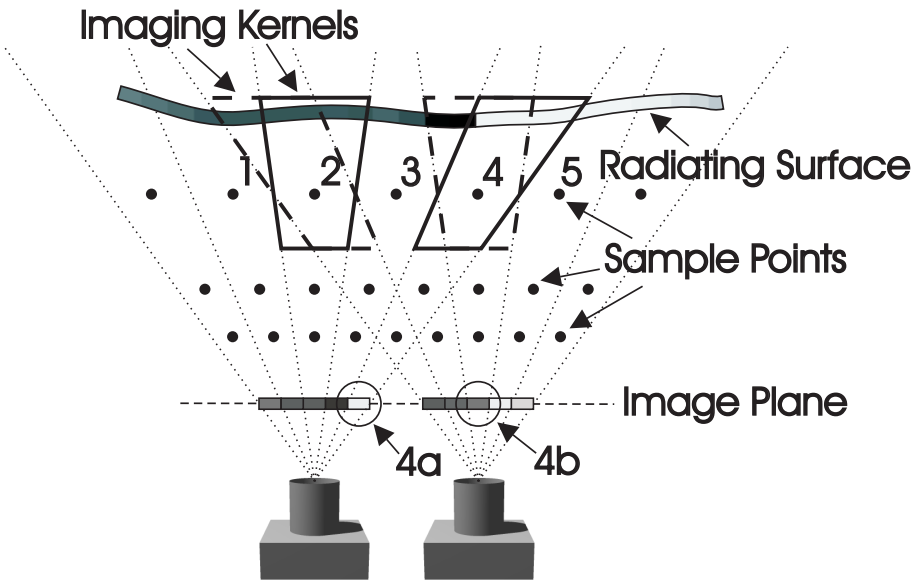


**Figure 2.9** Variations in the imaging convolution kernel between images can result in differences in the sampled radiances, $\overline{R}_i(x, y, Z)$, and transmittances, $\tilde{T}_i(x, y, Z)$. These differences may be significant at sample points, such as sample 4, where $\overline{R}_i(x, y, Z)$ varies greatly between neighbouring samples. This variation leads to differences in the observed pixel intensity between images of a common sample point, as demonstrated by comparing the observed intensity in pixels $4a$ and $4b$.

ditional computation, is that it complicates the optimisation process by increasing the interaction between sample points. In most cases, a reasonable result can be obtained by filtering if the bandwidth of the sampled scene is larger than the bandwidth of the imaging kernel.

An alternative approach is to position the cameras so that variation in the imaging kernels between cameras is minimised. If the differences in the imaging kernels are sufficiently small, filtering can be avoided altogether and any remaining differences between $\overline{R}_i(x, y, Z)$ and $\overline{R}_j(x, y, Z)$, and $\tilde{T}_i(x, y, Z)$ and $\tilde{T}_i(x, y, Z)$ simply treated as additional image noise. As shown in Fig. 2.9, these variations will be larger in regions where $\overline{R}_i(x, y, Z)$ and $\tilde{T}_i(x, y, Z)$ vary significantly between neighbouring samples. Therefore, an improved approach is to model the observed differences in $\overline{R}_i(x, y, Z)$ and $\tilde{T}_i(x, y, Z)$ between cameras, as a function of the intensity variations within each image. This is discussed in more detail in Section 6.4.

With a planar camera layout, variations in the convolution kernel $W_i(x, y, Z, u, v, w)$ between the cameras can be reduced by decreasing the width of the kernel in the $Z$ direction. This corresponds with an increase in the sampling resolution in this direction. For a more general camera layout, where the cameras surround the scene, variation in $W_i(x, y, Z, u, v, w)$ is minimised if $W_i(x, y, Z, u, v, w)$ is radially symmetric and all the cameras are approximately the same distance from a central reference point within the scene. This corresponds most closely with a uniform voxel spacing.

### 2.3.5 Transmittance

So far the image formation process has been described in terms of scene intensities and the transmittance of these intensities towards the various cameras. Such a model is not particularly useful by itself, since the transmittance properties depend on the camera positions. A more general scene model that is independent of camera position is usually required. To do this, the scene transmittances can be expressed in terms of per unit transmittances, as given by Eq. 2.12. This introduces an additional integral into the mapping and requires the per unit transmittances to be defined at every point within the scene. As with pixel ray integration, the system mapping can be simplified by approximating the low-pass filtered transmittance $\tilde{T}_i$ towards the $i^{\text{th}}$ camera, as a product of low-pass filtered regional transmittances.

From Eq. 2.12, and replacing $x, y, z$ with the dummy variables of integration $u, v, w$, the log transmittance towards the $i^{\text{th}}$ camera can be broken down into the summation of $k$ sub-integrals, giving

$$\begin{aligned}
\log(T_i(u,v,w)) &= \frac{1}{\cos\theta}\int_{w'=0}^{w} \log(\tau(u,v,w'))\,\mathrm{d}w' \\
&= \frac{1}{\cos\theta}\int_{w'=d_{i(k-1)}(x,y,u,v)}^{w} \log(\tau(u,v,w'))\,\mathrm{d}w' \\
&\quad + \sum_{n=1}^{k-1} \frac{1}{\cos\theta}\int_{w'=d_{i(n-1)}(x,y,u,v)}^{d_{in}(x,y,u,v)} \log(\tau(u,v,w'))\,\mathrm{d}w' \\
&= \log(T_{\omega i}(x,y,Z_{ik}(x,y),u,v,w) + \sum_{n=1}^{k-1} \log(T_{\nu i}(x,y,Z_{in}(x,y),u,v)),
\end{aligned} \tag{2.50}$$

where $Z_{in}(x,y) = (d_{in}(x,y,x,y) + d_{i(n-1)}(x,y,x,y))/2$ represents the $Z$ coordinates of each subregion, $T_{\omega i}(x,y,Z_{ik}(x,y),u,v,w)$ is the transmittance from point $(u,v,w)$ to $(u,v,d_{i(k-1)}(x,y,u,v))$, and $T_{\nu i}(x,y,Z_{in}(x,y),u,v)$ is the transmittance from point $(u,v,d_{in}(x,y,u,v))$ to $(u,v,d_{i(n-1)}(x,y,u,v))$. Taking exponents gives

$$T_i(u,v,w) = T_{\omega i}(x,y,Z_{ik}(x,y),u,v,w)\prod_{n=1}^{k-1} T_{\nu i}(x,y,Z_{in}(x,y),u,v). \tag{2.51}$$

Next, the transmittances $T_{\nu i}$ can be re-expressed as the sum of a low-pass filtered value in the $x$ and $y$ directions, plus a difference term, giving

$$T_{\nu i}(x,y,Z_{in}(x,y),u,v) = \overline{T}_{\nu i}(x,y,Z_{in}(x,y)) + \acute{T}_{\nu i}(x,y,Z_{in}(x,y),u,v), \tag{2.52}$$

where

$$\overline{T}_{\nu i}(x,y,Z_{in}(x,y)) = \frac{\iint_{-\infty}^{\infty} T_{\nu i}(x,y,Z_{in}(x,y),u,v)h_i(x,y,u,v)\,\mathrm{d}u\,\mathrm{d}v}{\iint_{-\infty}^{\infty} h_i(x,y,u,v)\,\mathrm{d}u\,\mathrm{d}v}. \tag{2.53}$$

Substituting Eq. 2.52 into Eq. 2.51 gives

$$\begin{aligned} T_i(u,v,w) = {} & T_{\omega i}(x,y,Z_{ik}(x,y),u,v,w) \\ & \times \prod_{n=1}^{k-1} \left( \overline{T}_{\nu i}(x,y,Z_{in}(x,y)) + \acute{T}_{\nu i}(x,y,Z_{in}(x,y),u,v) \right). \end{aligned} \tag{2.54}$$

Using vector variables $S = (x,y,z)$ and $t = (u,v,w)$, and using $\overline{T}_{\nu in}(x,y)$ and $\acute{T}_{\nu in}(x,y,u,v)$ to represent $\overline{T}_{\nu i}(x,y,Z_{in}(x,y))$ and $\acute{T}_{\nu i}(x,y,Z_{in}(x,y),u,v)$ respectively, Eq. 2.54 can be substituted into Eq. 2.33 and expanded to give

$$\begin{aligned} \tilde{T}_i(s) &= \frac{1}{\psi_{W_iK_i}} \int_{-\infty}^{\infty} T_{\omega i}(s,t) \prod_{n=1}^{k-1} \left( \overline{T}_{\nu in}(x,y) + \acute{T}_{\nu in}(x,y,u,v) \right) W_i(s,t)K_i(t)\,\mathrm{d}t \\ &= \frac{1}{\psi_{W_iK_i}} \int_{-\infty}^{\infty} T_{\omega i}(s,t)W_i(s,t)K_i(t) \prod_{n=1}^{k-1} \overline{T}_{\nu in}(x,y)\,\mathrm{d}t \\ &\quad + \frac{1}{\psi_{W_iK_i}} \int_{-\infty}^{\infty} T_{\omega i}(s,t)W_i(s,t)K_i(t)\acute{T}_{\nu i1}(x,y,u,v) \prod_{n=2}^{k-1} \overline{T}_{\nu in}(x,y)\,\mathrm{d}t \\ &\quad + \dots, \end{aligned} \tag{2.55}$$

where

$$\psi_{W_iK_i} = \int_{-\infty}^{\infty} W_i(s,t)K_i(t)\,\mathrm{d}t. \tag{2.56}$$

Now, assuming $\acute{T}_{\nu in}(x,y,u,v)$ is uncorrelated with the transmittances $\acute{T}_{\nu im}(x,y,u,v)$ and $\overline{T}_{\nu im}(x,y,u,v)$, for all $n \neq m$, the terms in Eq. 2.55 will all equal zero except the

first one. Therefore, the low-pass filtered scene transmittances can be expressed as

$$\begin{aligned}
&\tilde{T}_i(x,y,Z_{ik}(x,y)) \\
&\quad= \frac{1}{\psi_{W_iK_i}} \int_{-\infty}^{\infty} \Big[ T_{\omega i}(x,y,Z_{ik}(x,y),u,v,w) W_i(x,y,Z_{ik}(x,y),u,v,w) K_i(u,v,w) \\
&\qquad\qquad \times \prod_{n=1}^{k-1} \overline{T}_{\nu i}(x,y,Z_{in}(x,y)) \Big] \,\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}w \\
&\quad= \frac{1}{\psi_{W_iK_i}} \int_{-\infty}^{\infty} T_{\omega i}(x,y,Z_{ik}(x,y),u,v,w) W_i(x,y,Z_{ik}(x,y),u,v,w) K_i(u,v,w) \mathrm{d}u\mathrm{d}v\mathrm{d}w \\
&\qquad\qquad \times \prod_{n=1}^{k-1} \overline{T}_{\nu i}(x,y,Z_{in}(x,y)) \\
&\quad= \tilde{T}_{\omega i}(x,y,Z_{ik}(x,y)) \prod_{n=1}^{k-1} \overline{T}_{\nu i}(x,y,Z_{in}(x,y)),
\end{aligned} \tag{2.57}$$

where $\tilde{T}_{\omega i}(x,y,Z_{ik}(x,y))$ is the average weighted transmittance from radiating points within the region $\{u,v,w : W_i(x,y,Z_{ik}(x,y),u,v,w) > 0\}$ to the edge of that region, in the direction of the $i^{\text{th}}$ camera. This expresses the locally averaged or low-pass filtered scene transmittance towards the $i^{\text{th}}$ camera as a discrete product of average weighted or low-pass filtered regional transmittances. Substituting Eq. 2.57 into Eq. 2.49, the pixel intensities can be written as

$$C_i(x,y) = \sum_{k=1}^{K} \overline{R}_i(x,y,Z_{ik}(x,y)) \tilde{T}_{\omega i}(x,y,Z_{ik}(x,y)) \prod_{n=1}^{k-1} \overline{T}_{\nu i}(x,y,Z_{in}(x,y)) + N_i(x,y). \tag{2.58}$$

Finally, the expression for pixel intensities can be simplified by using $\overline{R}_{\nu i}(x,y,Z(x,y)) = \overline{R}_i(x,y,Z(x,y))\tilde{T}_{\omega i}(x,y,Z(x,y))$, to represent the average weighted radiance transmitted by the region $\{u,v,w : W_i(x,y,Z_{ik}(x,y),u,v,w) > 0\}$ in the direction of the $i^{\text{th}}$ camera, giving

**Theorem 3**

$$C_i(x,y) = \sum_{k=1}^{K} \overline{R}_{\nu i}(x,y,Z_{ik}(x,y)) \prod_{n=1}^{k-1} \overline{T}_{\nu i}(x,y,Z_{in}(x,y)) + N_i(x,y).$$

In most situations the expression for camera pixel intensities given in Theorem 3 is a reasonable approximation to the integral given in Theorem 1. If a better overall approximation to the pixel intensities is required, then the integral in Theorem 1 can be broken down into the summation of a number of sub integrals over $x$ and $y$. Each of these sub integrals is then given by Theorem 3, but with a narrower convolution kernel $W_i(x,y,Z,u,v,w)$ in the $u$ and $v$ directions.

## 2.4 RESOLUTION LIMITS

Given the image data, there is only a finite level of scene detail that can be accurately reconstructed. Therefore, it is pointless to represent or parameterise the scene at too high a resolution. The maximum scene resolution is governed by the bandwidth and sampling resolution of the image intensities $\overline{I}_i$, as well as the focal length and position of each camera.

From Theorem 1, the maximum bandwidth of $\overline{I}_i$ is simply equal to the bandwidth of the windowing function $h_i(x, y, u, v)$. For most well focused images this windowing function can be approximated by a smoothed spatially invariant rectangular window $h_i(x - u, y - v)$ whose width and breadth are equal to the sensor spacing. The -3 dB bandwidth of such a function is approximately $\frac{1}{2x_p}$ and $\frac{1}{2y_p}$ in the $x$ and $y$ directions respectively, where $x_p$ and $y_p$ are the $x$ and $y$ spacing between sensor elements. For blurred or out of focus images, the width of this windowing function will be greater, resulting in an even smaller bandwidth. This has the useful property that, in almost all instances, incident image intensities are sampled at or above the Nyquist rate. Consequently, the image resolution will be limited by the bandwidth of the windowing function rather than by the sampling resolution. In some situations it may be necessary to deliberately introduce a small amount of blurring to ensure this occurs. However, even with added blurring, a degree of aliasing will always be present, as any real windowing function will not filter out all high frequency components completely.

Given the image resolution, the resolvable resolution at any point within the scene can be found by multiplying the image resolution by the projection magnification factor at that point. This magnification factor $\kappa(X, Y, Z, \phi)$ is defined as the ratio of projected image length $\delta l$ to actual scene length $\delta L$ for an infinitesimally small line segment passing through the scene point $(X, Y, Z)$ at an angle $\phi$ to the image plane. Using the sine rule and trigonometric identities $\sin(90 + x) = \cos x$ and $\cos(-x) = \cos x$, the magnification factor is given by

$$\kappa(X, Y, Z, \phi) = \frac{d_i \cos(\theta - \phi)}{Z \cos\theta}, \tag{2.59}$$

where $\theta = \arctan(\sqrt{X^2 + Y^2}/Z)$ is the angle between the optical axis and a ray passing through the principal point of the camera and the scene point (see Fig. 2.10). Therefore, the maximum resolvable scene resolution is $\frac{d_i}{Zx_p}$ and $\frac{d_i}{Zy_p}$ in the $X$ and $Y$ directions and zero or unresolvable along any line passing through the optical centre of the camera. Consequently, scene resolution is highest close to the camera in a direction perpendicular to the pixel rays and zero along any pixel ray.

With multiple cameras, determination of the maximum resolvable resolution becomes rather more complicated. In certain situations, the effective image sample spacing can be reduced by adding more cameras. However, assuming the image resolution is limited by the optics and sensor width, rather than the sample spacing, this will not improve the image resolution. In this case, the resolvable scene resolution at any

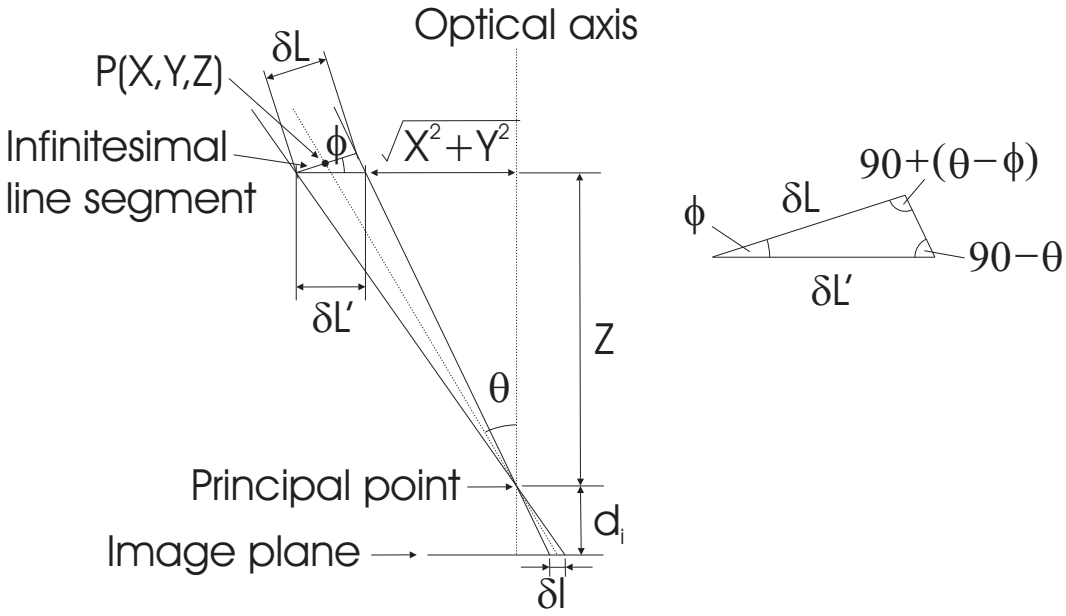


**Figure 2.10** The projection magnification factor of the imaging system, $\kappa(X, Y, Z, \phi)$, is defined as the ratio of projected image length $\delta l$ to actual scene length $\delta L$ for an infinitesimally small line segment centred at scene point $(X, Y, Z)$ and at an angle $\phi$ to the image plane. Using the sine rule, the actual scene length $\delta L$, can be related to the apparent scene length $\delta \acute{L}$, by $\delta L = \delta \acute{L} \sin(90 - \theta)/\sin(90 + (\theta - \phi))$. This can be expressed in terms of $\delta l$, using $\delta \acute{L} = \delta l Z/d_i$. Using the trigonometric identities $\sin(90 + x) = \cos x$ and $\cos(-x) = \cos x$, the projection magnification factor can be expressed as $\kappa(X, Y, Z, \phi) = d_i \cos(\theta - \phi)/Z \cos\theta$.

point is simply equal to the maximum resolution that can be resolved by any of the individual cameras at that point. Therefore, with two or more cameras, it is possible to infer information or detail about the scene along any pixel ray, so long as at least one of the cameras can resolve detail along this line. The maximum resolvable scene resolution can be changed by altering the position of each camera. Accordingly, choice of camera positioning is important and will vary depending on the application.

## 2.5 SCENE SAMPLING

With the majority of scene models, including most continuous representations, various properties of the scene are evaluated at a finite number of discrete points during the reconstruction process. With a discrete volumetric model, these sample points usually correspond with the set of scene parameters, while with a depth-map or continuous model, data at these points are used to estimate the scene parameters. In both cases, an appropriate set of sample points should be used to obtain the best results.

The choice of sample points depends on the application and scene resolution. A sample spacing and associated convolution kernel that is closer than the resolvable scene resolution leads to further ambiguities in the inverse mapping, since the defined scene model will be inadequately sampled by the camera images. It also increases

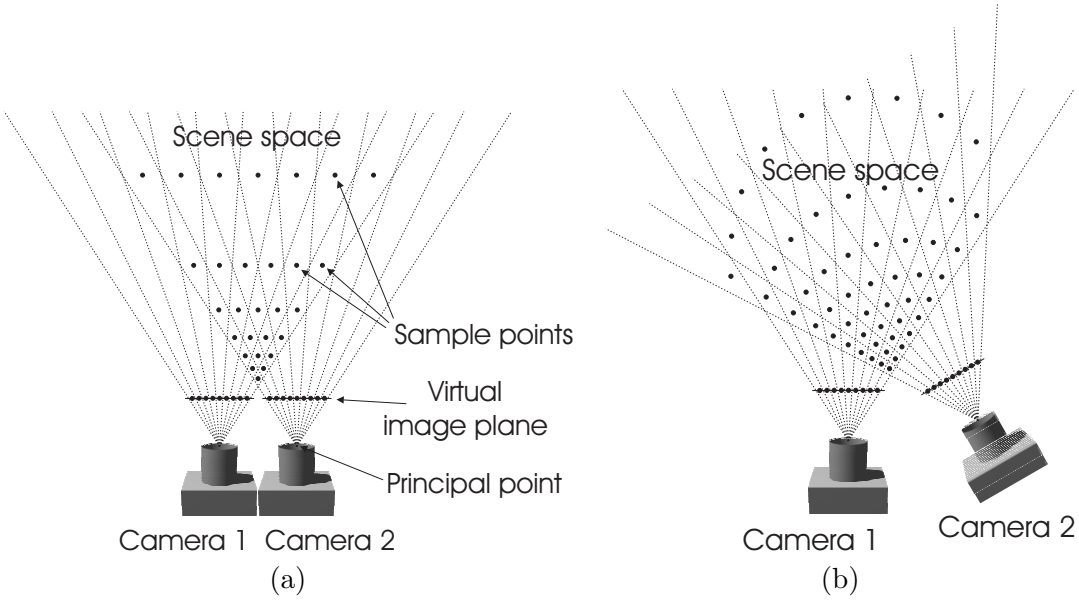


**Figure 2.11** Minimum resolvable sample spacing for a two camera system. (a) Planar camera system, where both cameras lie on the same plane and face perpendicular to that plane. With this configuration, sample points coincide with planes that are uniformly spaced in inverse depth. (b) Arbitrary two camera system.

the computational complexity because of the extra samples that are required to be estimated. On the other hand, under sampling the scene results in an ambiguous forward mapping and lower reconstructed resolution.

For a planar two camera system, the minimum resolvable sample spacing is shown in Fig. 2.11(a). As observed, the sample points coincide with planes that are uniformly spaced in inverse depth. Within each plane, the sample spacing is proportional to the depth of the plane from the camera plane. Such a configuration is referred to as integer disparity sampling, as the spacing between samples along a given pixel ray corresponds to integer shifts in pixel disparity between the two images. This sampling scheme can be applied to any two camera system as shown in Fig. 2.11(b), although sample points will no longer correspond with planes of constant depth.

One of the big advantages of integer disparity sampling is that sample positions correspond exactly with pixel locations when projected onto the image plane of each camera. This simplifies the mapping from scene parameters to image data, since the discrete mapping points, $\{(x, y, Z_{ik}(x, y)) : x, y, k \in \mathbb{Z}\}$, and corresponding convolution kernel, $W_i(x, y, Z_{ik}(x, y), u, v, w)$, implicit in Theorem 3, can be chosen to coincide with the set of sample points. Therefore, no interpolation or averaging is needed when transforming from one parameter set to the other. It also simplifies the process of determining a point's visibility, as only those points which lie along the intersecting rays need be considered. Unfortunately in general, this sampling scheme is only applicable to two camera systems, as the intersection between pixel rays from multiple cameras will not necessarily coincide.

When using more than two cameras, the resolvable scene resolution will vary somewhat irregularly throughout the scene depending on which cameras can observe a particular region. Consequently, it is impossible to select a regular set of points and associated convolution kernels that match the resolvable scene resolution. In most situations it is also impossible to position the samples so that they correspond exactly with pixel rays from each camera. Therefore, some form of interpolation must be used to map between the scene samples and the image data.

For planar camera configurations, two common sampling schemes are used. These both involve uniform sampling under a disparity coordinate system. In the first approach, sample points are positioned so as to correspond to an integer disparity shift between pixels in neighbouring cameras. Such a sampling scheme is shown in Fig. 2.12(a). If the cameras are evenly spaced on a regular grid then the sample points will correspond exactly with pixel ray intersections. As with integer disparity sampling for two cameras, this allows the discrete mapping points for each image to correspond exactly with the set of scene sample points. Unfortunately, to prevent aliasing, the convolution kernel, $W_i(x, y, Z, u, v, w)$, must be elongated in the $Z$ direction so as to match the sample spacing. This results in large variations in the kernel shape between images. If a narrower kernel in the $Z$ direction is used instead to reduce variation between images, then the scene will be inadequately sampled. Another consequence of this sampling scheme is that the resolution in the $Z$ direction is significantly less than the resolvable limits of the system in many places.

An alternative approach is to arrange the samples so as to correspond to an integer disparity shift between the pixels in the two outermost cameras. This results in a finer spacing between samples in the $Z$ direction, as shown in Fig. 2.12(b). However, some form of interpolation is required to map between the sample points and the discrete mapping points, $\{(x, y, Z_{ik}(x, y)) : x, y, k \in \mathbb{Z}\}$. Within the region that is visible to all cameras, the resulting sampling resolution is equal to the maximum resolvable scene resolution. Outside this region, the sample spacing is closer than what can be resolved. In most situations this is not a problem, except that more samples are used than is necessary. With a number of stereo algorithms, this region is outside the defined scene volume and so can be ignored anyway.

To deal with more general camera systems or provide an arbitrary scene resolution, a variety of other sampling schemes can be used. These can be useful in certain situations, but usually result in a more complex mapping between the scene and image parameters. One such approach is to position samples on a regular $X, Y, Z$ grid throughout the scene volume [Seitz and Dyer 1999, Culbertson et al. 1999, Slabaugh et al. 2000b]. This is useful in some applications where the scene is constrained to lie within a known finite volume, and must modelled at a fixed resolution. An alternative approach presented by Slabaugh et al. [2000a] is to warp voxels based on their position within some user-defined voxel space. This allows the scene to be sampled independently of camera position or

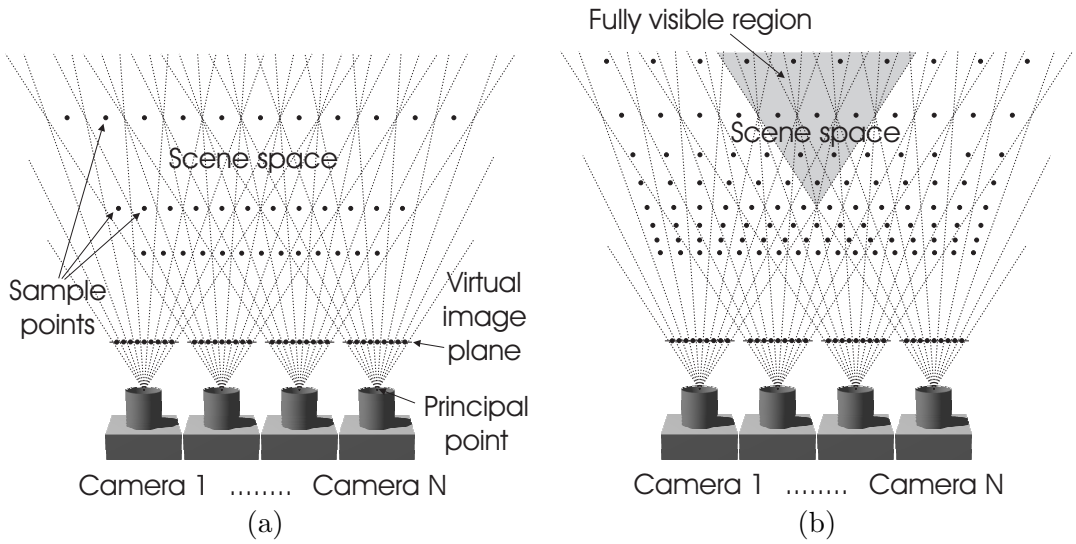


**Figure 2.12** Planar multiple camera sampling schemes. (a)Nearest neighbour integer disparity sampling, where sample points are positioned so as to correspond with integer disparity shifts between neighbouring cameras. (b) Furthest neighbour integer disparity sampling, where sample points are positioned so as to correspond with integer disparity shifts between the two outermost cameras.

resolution and enables the reconstruction volume to accommodate a semi-infinite or infinite region. Environment mapping [Greene 1986] can also be used to deal with large or infinite scenes. This approach is commonly used in the computer graphics domain, where background or distant objects are represented by a texture mapped sphere or cube that surrounds the foreground scene. Although convincing synthetic images can be produced, this method is inappropriate for most scene reconstruction applications as the three-dimensionality of the background is lost. It also requires separate modelling of the foreground and background, leading to difficulties in the reconstruction process.

## 2.6 PRIOR KNOWLEDGE

To improve the estimation process, prior knowledge about a scene can be incorporated into the reconstruction process. This allows additional bits of information to be used that are not available from the camera images. The most common way of doing this is to apply hard constraints to the set of model parameters. This is the simplest approach. It limits the range of possible scene estimates, hopefully reducing the probability of a poor reconstruction. Such constraints can be applied to individual parameters or on the allowable combination of parameter values. For example, the continuity constraint enforces opaque surfaces to be linked together so that they form a continuous surface.

In addition to imposing hard constraints, prior statistical information relating to the scene parameters can be applied. This is a more general and flexible approach, as both hard constraints and arbitrary probabilistic information can be used. For instance, the scene will usually consist of cohesive opaque and transparent regions rather than a random cloud of points. Therefore, preference should be given to neighbouring

samples that have the same opacity value, although differences should still be allowed. Such statistical information can be applied by modifying the overall joint probability distribution of the model so as to reflect the prior information.

There are many examples of prior information that can be used to aid the reconstruction process. These range from general priors that are valid for most scenes, to more specific information that is only applicable in a few select cases. An important requirement with all of these is that the information is applicable to the problem at hand. Although potentially useful, care must be taken when applying such information, as the inclusion of invalid priors can severely degrade the scene reconstruction. Because of the large effect prior information has on reconstruction performance, choice and implementation of appropriate priors is a vital component of the system model and reconstruction process. The following subsections outline some of the more common priors that can be applied to the reconstruction problem.

### 2.6.1 Object opacity

Constraints and prior information relating to the scene transmittances or opacities are used by most stereo algorithms. If applicable, these constraints are extremely useful, as they can be used to considerably simplify the mapping between scene parameters and pixel intensities.

There are four key assumptions relating to scene opacity that are commonly applied to the scene reconstruction problem. The first is that the local scene transmittances, $T_{\nu i}$, are either 0 or 1, corresponding to fully opaque or transparent media respectively. The second is that the transmittance through any reflecting or light emitting region is zero. The third assumption is that the local scene transmittances, $T_{\nu i}$, are highly correlated. The final assumption is that the average local transmittance through any windowed region varies smoothly with viewing direction. In most situations these priors are approximately true, as the scene typically consists of a number of cohesive opaque objects located within a transparent medium, usually air. An exception to this is when the scene contains glass objects or windows, which both reflect and transmit light. Another notable exception is when the scene contains regions consisting of a cloud of opaque surfaces, such as leaves on a distant tree. In this case, neighbouring points are likely to have different transmittances, and the average weighted transmittance through a windowed region will be somewhere between zero and one.

For situations where the region defined by the convolution kernel $W_i$ contains an opaque boundary or surface, the average weighted transmittance through the region will be zero, so long as the surface extends fully across the region in the $x, y$ directions. This will not always be the case, as shown in Fig. 2.13(a). However in most instances, the scene can be reasonably well approximated by a set of fully opaque or transparent samples (see Fig. 2.13(b)). As shown, for steeply sloping surfaces, or depth discontinuities,

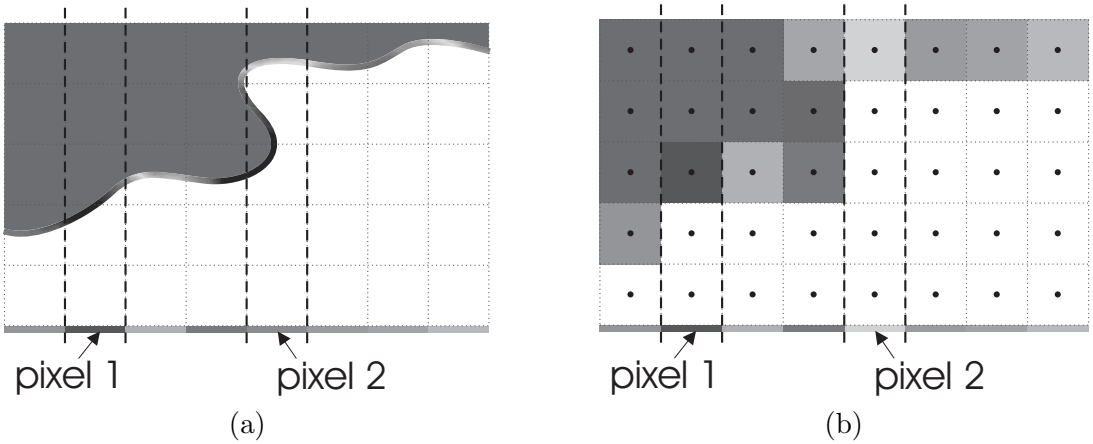


**Figure 2.13** (a) A section of the scene, consisting of an opaque radiating surface, is observed from below by one of the system cameras. The recorded pixel intensities are shown along the bottom of the figure. This is modelled using a set of samples that are obtained after convolving the scene with the imaging convolution kernel $W_i$. In most situations, these samples can be closely approximated by a set of samples that are either completely transparent or opaque (b). For clarity, the windowed region defined by $W_i$ for each sample, is shown as a dashed square. For viewing angles close to the surface normal, the binary transmittance approximation is reasonably accurate, and the modelled pixel intensities will be close to what is observed. This can be seen be comparing the intensities at pixel 1, between the approximated and actual scene. However, for steeply sloping surfaces, this approximation can lead to significantly different values, as shown with pixel 2.

this approximation is rather poor and can lead to large variations in the predicted pixel intensities. Therefore, if using this constraint, depth discontinuities or steeply sloping surfaces should be detected and dealt with appropriately.

By making the approximation that the average weighted transmittance though any sampled windowed region is fully transparent or opaque independent of viewing direction, the regional transmittances can be expressed as $\overline{T}_\nu$, where $\overline{T}_{\nu i} \approx \overline{T}_\nu$ for all camera images. If it is also assumed that the transmitted radiance from all transparent regions is zero, each pixel will observe at most one radiating region. Therefore, the expression for pixel intensities in Theorem 3 can be simplified to give

**Theorem 4**

$$C_i(x,y) = \begin{cases} \overline{R}_{\nu i}(x,y,Z_i^*(x,y,\overline{T}_\nu)) + N_i(x,y) & \textit{if } Z_i^*(x,y,\overline{T}_\nu) \textit{ exists} \\ \overline{R}_{Bi}(x,y) + N_i(x,y) & \textit{otherwise,} \end{cases}$$

where $Z_i^*(x,y,\overline{T}_\nu)$ is the discrete depth of the nearest opaque region intercepted by the pixel ray $(x,y)$, and $\overline{R}_{Bi}(x,y)$ is the average background radiance along that pixel ray. If there is no opaque region along a pixel ray, then $C_i(x,y)$ will simply equal the average background radiance, $\overline{R}_{Bi}(x,y)$, plus image noise. Using this simplification, pixel intensities will be equal to the radiance of a single sample point, rather than the integration, or summation, of radiances along a line. This is the basis of the discrete depth-map model, where the objective is to determine the depths $Z_i(x,y) = Z_i^*(x,y,\overline{T}_\nu)$.

This contrasts with medical imaging tomography where all points along the ray corresponding to an image point will contribute to its intensity. In this case, the objective is to estimate an object's density as a function of spatial position by measuring the amount of radiation it absorbs in various directions. By replacing absorption with radiance, this is equivalent to the scene reconstruction problem with unity transmittance throughout the scene.

In many instances, prior information about scene opacities is applied using a number of related constraints. Although these are based on the opacity priors, such constraints are often applied in combination with additional assumptions, or simplifications to the system mapping. Consequently, although these constraints usually help to constrain the optimisation process, the resulting reconstruction is often degraded due to inaccurate modelling.

In the simplest case, assuming Lambertian reflection, the opacity constraints imply that if a small surface region is visible in two or more cameras, then the pixel intensities corresponding to that region will be similar. This forms the basis of most stereo algorithms, which attempt to identify surface regions through the similarity in their projected intensities. With traditional stereo matching this is achieved by matching pixels between pairs of images and then using simple triangulation to determine the spatial position of the corresponding surface. However, due to noise and occlusions this process is often ambiguous, and in some cases a match will not exist. With the majority of algorithms this leads to inconsistencies, as the matching is performed independently of the visibility interaction between scene regions. Consequently, the resulting scene reconstruction is often far from optimal, especially for scenes containing numerous occlusions.

An improved approach is to implement the so called uniqueness constraint. This was first proposed by Marr and Poggio [1976] for matching binary image features and is based on the idea that each image feature point must correspond to a unique point in scene space. The uniqueness constraint states that each point in one image should match at most one point in the other image. This implies that only one match can exist along any line of sight. To enforce this constraint Marr and Poggio [1976] propose a cooperative algorithm where matches along the same line of sight inhibit each other. The algorithm was later modified by Zitnick and Kanade [1999] to work with greyscale images. However, both these algorithms fail to reconstruct scene points which cannot be matched in both images. They are also unsuitable for systems containing more than two cameras.

Instead of applying related constraints to various scene parameters or mapping functions, the opacity constraints can be applied directly to the scene parameters. To do this, a volumetric model of the scene must be used, where opacity is explicitly represented. This is the most general approach, allowing the visibility interaction between regions to be more accurately modelled.

### 2.6.2 Surface continuity

The assumption that local scene transmittances, $T_{\nu i}$, are correlated, can be extended to the average weighted regional transmittances, $\overline{T}_{\nu i}$. Under this assumption, neighbouring regions are likely to have the same or similar opacity as each other. Assuming binary regional transmittances, this can be expressed in terms of surface continuity, where opaque regions are likely to be grouped together, forming several piecewise continuous surfaces, rather than a cloud of disjoint regions. With a depth-map model, this assumption implies that neighbouring surface points are likely to have the same or similar depth. In some situations, this assumption can be extended, by constraining the scene be a single continuous surface [Gimel'farb 1998]. Although usually untrue, this is a good approximation for aerial photogrammetry, where the terrain surface is viewed from a significant height above the surface.

A variation of the surface continuity assumption is the ordering constraint, where the ordering of points is assumed to be preserved between images [Cox et al. 1996, Gimel'farb 1998]. In this case, if a point $m$ is observed to the left of a point $n$ in one image it should appear to the left in all other images, so long as it is visible. This constraint is usually used with more traditional stereo algorithms, to help reduce false matches. For situations where the cameras are located close together and face in a similar direction, this constraint is usually true. However, it does not apply when the scene contains, for example, small opaque regions located in front of a distant surface.

### 2.6.3 Surface smoothness

Furthering the idea of surface continuity, priors relating to surface smoothness can also be used. These are assumptions that relate to the change in slope, or curvature, between adjacent surface regions. In most instances, the change in average slope between neighbouring regions will be small. Therefore, preference should be given to surfaces which appear relatively smooth after being convolved with the imaging kernel. Although smoothness priors are applicable to most scenes, they are often poorly implemented. In most cases, the difference in depth between adjacent samples is minimised rather than the change in slope. This is usually done because it reduces the computation time and simplifies the optimisation process. Calculating a change in depth requires only two points to be compared, whereas calculating a change in slope requires at least three. This tends to favour fronto-planar surfaces, as these correspond to a minimum change in depth. Another common mistake is to assume that small image regions correspond to areas of approximately constant depth [Faugeras et al. 1993, Fusiello et al. 1997, Kanade and Okutomi 1994]. Although a reasonable assumption for certain scenes and camera configurations, it is invalid at any apparent object boundaries, where large depth discontinuities can occur. Surface smoothness can also be implemented as a gradient limit constraint, where the change in depth between adjacent pixels is

constrained to be less than some threshold.

### 2.6.4 Visibility assumptions

The difficulty with calculating a point's visibility or transmittance towards a camera, using Eq. 2.57, is that it depends on the opacity of numerous other regions within the scene. Consequently, estimation of the scene parameters must be considered as a whole, rather than as a number of isolated points or surfaces. This complicates the estimation process, as the resulting reconstruction function is hard to optimise, due to the high level of interaction between scene parameters.

Rather than modelling the overall transmittance or visibility of a point as a function of regional transmittances, as given by Eq. 2.57, the visibility of a region can be inferred directly from the image data or from prior knowledge. This is a less accurate way of modelling the system but does reduce the interaction between scene parameters. Therefore, the resulting model will be considerably simpler and easier to optimise, especially when the visibility assumptions are applied in conjunction with the regional binary opacity constraint. Even in situations where the visibility of each region is inferred directly from scene opacities, information about the likelihood of scene visibilities is useful, since it can help guide the reconstruction process. Techniques for dealing with visibility interactions are discussed in Section 3.5.

### 2.6.5 Lambertian reflectance

In addition to priors relating to the scene opacities, information or assumptions about the scene radiances can be used to improve and simplify the reconstruction process. The key assumption made by most reconstruction algorithms is that the radiance of a given scene point will vary smoothly with viewing angle. In most cases, this assumption can be extended to the average transmitted radiance within regions of the scene. Consequently, assuming the overall transmittance is the same in each direction, a radiating region will appear similar in different cameras, so long as the cameras are located close to one another. For scenes consisting of a number of reflecting surfaces, this assumption is based on the fact that most surfaces are reasonably matte and have a diffuse reflectance component that is significantly larger than the specular one.

With most scene reconstruction algorithms, the assumption that scene radiances vary smoothly with viewing angle is taken one step further by assuming the average radiance emitted from any region is equal in all directions. This allows the sampled radiances to be expressed as $\overline{R}_\nu$, where $\overline{R}_{\nu i} \approx \overline{R}_\nu$ for all camera images. For reflecting surfaces, this is equivalent to the Lambertian assumption, where the reflected intensity from a surface is independent of the viewing angle. For many scenes this is a reasonable approximation, although problems will occur when reconstructing shiny objects such as those made from plastics or metal. The advantage of Lambertian reflectance is that if

several cameras can observe the same region, the observed intensities will all be the same. This considerably simplifies the reconstruction process as the image data from different cameras can easily be compared, independent of surface angle, viewing direction, or the position of the light source.

### 2.6.6 Intensity and colour correlation

The other important prior relating to scene radiances that can be applied to the scene model is that the scene radiances are likely to vary piecewise smoothly across any surface. Although seldom applied directly, this prior can be used to help detect depth discontinuities or segment the images into surface patches. Because the radiances across different surfaces are usually independent of one another, a change of surface will usually correspond with a sharp change in the observed image intensity or colour. Therefore, depth discontinuities are most likely to correspond with a distinct change in observed image intensity or colour. This is important, as determining the boundary of surfaces is one of the trickiest parts of the reconstruction process and is essential for dealing with occlusion.

# Chapter 3

## RECONSTRUCTION TECHNIQUES

In Chapter 2 modelling of the scene and imaging was discussed. The system model describes the formation of images given a particular scene. This is a forward transform from a set of scene parameters to the image data. The scene reconstruction problem is the inverse of this, inferring the scene parameters from the image data.

Obtaining a reliable estimate of the scene from the image data is difficult because of the complex interaction between the scene parameters and the image data and because of the loss of information that occurs in the imaging process. Consequently a wide variety of techniques have been proposed for solving this problem. This chapter presents an overview of these techniques and discusses some of the current advances in this field.

The problem of determining the opacity and radiance of the scene given the image data is inherently difficult. These difficulties primarily arise because of the loss of information that occurs when projecting a 3D scene onto a 2D set of images. The problem is further complicated by system noise which adds a degree of uncertainty into the forward mapping. This leads to ambiguities in the inverse problem, since a large number of potential scenes could correspond with the observed data. The other complication is that the relationship between the scene and the images is complex and involves a high level of interaction between the various parameters.

Because of ambiguities in the inverse problem some form of regularisation is required to obtain a meaningful solution. In the absence of any additional information a common approach is to find a solution which both fits the data and minimises a pre-defined energy function. In the work described in this thesis a statistical approach is used, where the objective is to optimise the scene with regard to some statistical measure. Two commonly used measures are determining the scene with the Maximum A Posteriori (MAP) probability or finding the scene with the Minimum Mean Square Error (MMSE).

Section 3.1 of this chapter discusses MAP and MMSE estimation. Reconstruction techniques for scene reconstruction are introduced in Section 3.2, focusing on traditional approaches. A variety of common matching problems are discussed in Section 3.3. Following this, the incorporation of smoothness priors is considered in Section 3.4. Section 3.5 discusses various techniques for dealing with the difficult problem of scene

occlusions. Finally, a variety of commonly used techniques for function optimisation that have been applied to the scene reconstruction problem are presented Section 3.6.

## 3.1 BAYESIAN INFERENCE

A statistical approach allows ambiguities in the inverse problem to be dealt with objectively. Using this approach the scene reconstruction problem can be expressed as an optimisation problem, where the objective is to optimise the scene with respect to a given statistical measure. In most cases this is taken to be the scene which maximises the joint posterior distribution of the system (MAP), or which minimises the expected square error (MMSE). "System" here defines the scene and its relationship to the image data, as well as any prior information.

In many instances, exact statistics about prior information are usually unknown. In this situation, a Bayesian approach is adopted where prior statistics are assumed or approximately modelled to help improve the scene estimate. For a meaningful optimisation to be obtained, it is important that the applied priors are valid.

The choice of statistical measure can greatly affect the resulting scene estimate depending on the posterior distribution of the system. The commonly used MAP estimate gives the most likely estimate of the scene, given the available data and any prior information. This corresponds to the peak in the posterior distribution. In many cases, although this is the most likely estimate, it is unlikely to correspond exactly with the actual scene. In fact, using standard error metrics such as sum of absolute differences or sum of square differences over the set of scene parameters, the MAP estimate is unlikely in some situations to even be close to the actual scene.

The MMSE is another popular estimate used for obtaining solutions to inverse problems in the presence of uncertainties. The MMSE estimate corresponds to the mean of the posterior distribution. This estimate minimises the expected square error of the resulting reconstruction. One of the biggest problems with the MMSE estimate is that the resulting solution is likely to be infeasible when applied to a discrete probability distribution. This is a significant problem when dealing with binary opacities. Another major problem is that an error or distance metric must be defined over the range of scene states. With mixed distributions, such as opacity and radiance, this is hard to define. For example, it is difficult to say whether opaque black is further from opaque white than it is from transparent black. Also minimising the expected error for each variable independently may not minimise the overall expected error, depending on how the error is defined.

To highlight the differences between these two estimates, consider the posterior distribution shown in Fig. 3.1. In this example the map estimate returns the single most likely estimate. However, the expected square error of the estimate is high because the narrow peak sits away from the bulk of the distribution. Conversely, the MMSE

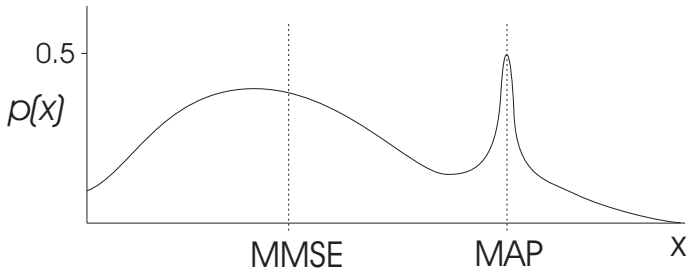


**Figure 3.1** The MAP estimate is shown to return the single most likely estimate. However, the expected square error of the estimate is high because the peak is narrow and away from the bulk of the distribution. This contrasts with the MMSE estimate which minimises the expected square error and corresponds with the mean of the distribution.

corresponds to the mean of the distribution. Although the probability of this exact estimate is less than the MAP estimate, it is likely to be closer to the true data than the MAP estimate, using a common error metric such as sum of square differences. In this example the two measures are different, however, in a lot of situations (such as a Gaussian shaped distribution) the two estimates will be similar.

Because of the differences between the MAP and the MMSE, one measure is usually more suited to a particular application than the other. In general the MMSE estimate is better suited to continuous distributions, such as estimating depths in a depth-map model, while the MAP estimate is better for discrete or mixed models, such as most volumetric models. Because this thesis focuses on discrete volumetric models, the MAP estimate is used in Chapters 4, 5 and 6.

## 3.2 RECONSTRUCTION TECHNIQUES

Given a statistical measure of the scene estimate, the scene reconstruction problem can be expressed as an optimisation problem where the objective is to maximise or minimise a function, subject to any of the imposed constraints. The problem is computationally extremely difficult and approximate solutions are sought because of the large solution spaces that are typically involved. For the scene reconstruction problem the posterior distribution is a complicated function of many variables. Although calculating the posterior distribution for a given scene is usually straightforward, obtaining the maximum or mean over all possible scenes is very difficult.

In principle this problem can be solved by searching through all possible combinations to find the most likely one. However, even for very small systems this approach is usually infeasible, since the number of combinations is enormous. For example, assuming a typical voxel model with dimensions $580 \times 300 \times 36$, where each voxel represents the binary opacities within the scene, the number of possible scenes is $2^{580\times300\times36} \approx 10^{1,800,000}$. Therefore, a direct or brute force approach is out of the question.

Rather than randomly evaluating potential scenes, the usual approach is to use

heuristic methods to find near optimal solutions to the objective function. The other approach is to simplify the objective function, so that optimisation becomes simpler. This is usually achieved by making a number of approximations or assumptions about the scene to simplify the system model. In practice, a combination of these two approaches is usually used.

### 3.2.1 Stereo matching

The problem of reconstructing a three dimensional scene from several viewpoints was first investigated in the fields of aerial photography and human stereopsis. Until relatively recently, the scene reconstruction problem was typically treated as a matching problem where the objective was to match points or features between two or more images. Having obtained a match, the three dimensional position of a point could be determined by triangulation assuming the camera positions were known.

The matching of image points is performed by comparing a region in one image, referred to as the reference image, with potential matching regions in the other image and selecting the most likely match based on some similarity measure. The resulting scene estimate is then invariably represented using a depth-map relative to the reference camera.

As an example of the stereo matching process, consider estimating the three dimensional position of a point $P$ shown in Fig. 3.2. By correctly matching this point between the two images, the relative shift or displacement of the point can be used to calculate the depth of the point. If all cameras are parallel and located on the same plane, the magnitude of this displacement or disparity $d$ is related to the depth $Z$ by

$$Z = \frac{Bd_i}{d}, \tag{3.1}$$

where $B$ is the baseline distance between two cameras and $d_i$ is the distance of the image plane behind the principal point. One problem with this approach is that it is difficult to determine matches reliably because of ambiguities and occlusions. To reduce the number of ambiguities, regions in the image are matched in order to improve the reliability of matching, instead of individual pixels. This is based on the assumption that nearby pixels are likely to have originated from a similar depth. However, difficulties arise in regions which do contain several depths, because the observed region will appear different between the various cameras. The spatial resolution of the reconstructed scene will also be reduced in proportion to the size of the matching region used.

Another difficulty with traditional stereo matching is which surfaces that are visible within the reference image may be occluded or hidden from view in one or more of the other images. In this situation false matches will occur as a true match does not exist. To avoid these problems occluded regions must be identified. Matches must then only be formed with images where the corresponding surfaces are visible. Identifying these

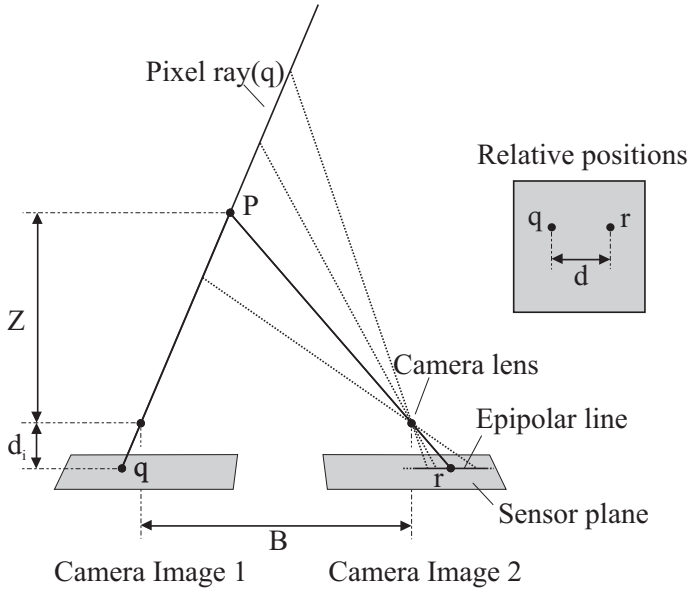


**Figure 3.2** Demonstration of disparity.

surfaces is difficult with traditional stereo matching, since the matching is performed directly in 2D image space where occlusions cannot be properly modelled.

### 3.2.2 Multiple camera stereo matching

The binocular stereo matching process can be extended to multiple images by selecting a common base or reference image and then matching regions in this image to all the other images. To do this, matches can be formed within each image separately and then combined to give an overall depth estimate. Alternatively, the depth of regions within the reference image can be obtained by matching all the other images simultaneously.

The use of multiple cameras [Gruen and Baltsavias 1988, Okutomi and Kanade 1993, Kanade et al. 1996] offers many advantages over standard two camera stereo. Ambiguous matches are often eliminated, the effects of noise are reduced and regions which are hidden or occluded in one image are often visible from other locations allowing a valid match to be made.

Although the use of multiple cameras helps improve results, the interactions between scene points cannot be modelled accurately in 2D. As a consequence, the basic stereo matching approach is only applicable to problems where there are a small number of occlusions between images. For certain setups, such as aerial photography, this is a reasonable assumption as the ground surface is usually smooth and continuous. It is also a good approximation for scenes that are imaged by closely spaced cameras, as the percentage of occlusions within each image is small.

A final problem with the traditional stereo matching approach is that it performs a local minimisation to find the best match of a point, rather than performing global

minimisation to find the most likely estimate of the scene. This results in a non-optimal as well as incomplete scene reconstruction.

### 3.2.3 Reference camera minimisation

To generalise the stereo matching approach, the matching process can be reformulated as selecting the best match along rays in space corresponding to pixels in the reference camera. For each point in the scene volume, a similarity measure is calculated by comparing the corresponding projected intensities in each image. The process of stereo matching then becomes equivalent to finding the best match along each pixel ray.

With traditional stereo matching, the similarity measures are calculated by comparing the projected reference camera intensities with the projected intensities in each of the other cameras. This places additional importance on the reference image, since any noise or errors in the reference image will have a strong effect on the estimate regardless of how many other cameras are in place.

To use the data in all images equally, a 'virtual camera' can be used where scene radiances are taken to be the average of the observed image intensities rather than equal to the reference intensities. This has the added advantage that the minimisation can be performed independently of the camera positions, as the virtual camera can be placed anywhere. However, calculating scene radiances in this way introduces a number of problems, since the visibility of a point must be known to determine its radiance.

In traditional stereo matching approaches it is assumed that the majority of surface points within a region of interest are visible in all images. The resulting modelling errors are simply treated as image noise or outliers. Even without these problems the reference camera minimisation approach is sub-optimal as the minimisation is only performed with respect to one camera. This results in an incomplete reconstruction that often poorly corresponds with the remaining views.

### 3.2.4 Global optimisation

To improve results the scene reconstruction problem can alternatively be approached from a global perspective, where the objective is to directly minimise or maximise the scene reconstruction objective function. Traditionally this approach was considered too computationally intensive. However, with improvements in computing speed and the recent development of efficient global optimisation techniques such as graph cuts [Kolmogorov et al. 2003, Bleyer and Gelautz 2007, Lin and Tomasi 2004, Tsin et al. 2003, Tran and Davis 2006, Hong and Chen 2004] and belief propagation [Sun et al. 2002, Sun et al. 2003, Sun et al. 2005], this approach has proved very successful. Almost all of the current top performing algorithms are now based on global optimisation techniques [Szeliski et al. 2006]. A variety of techniques for performing global optimisation are presented in Section 3.6.

## 3.3 MATCHING PROBLEMS

Most scene reconstruction algorithms rely on the property that regions or features appear similar from different camera positions. Although usually true, variations in the observed intensity of a region do occur and lead to a number of difficulties. This problem is compounded by the fact that the system model is only an approximation to the true system and introduces additional errors. Intensity variations can be grouped into four categories: variations that are caused by the cameras or inaccurate modelling of the cameras; variations caused by differences in the region of integration for each pixel between the cameras; variations attributed to specular reflections and other lighting variations between the images; and variations due to system noise.

### 3.3.1 Camera calibration

A common cause of intensity variation between images of a common region are camera distortions. These distortions can affect both the position and intensity of the incident light recorded by the camera.

An ideal pinhole camera model is usually assumed to simplify the mapping between scene and image parameters. However, in many cases this is a rather poor approximation to actual cameras and the recorded image intensities often vary from those predicted [Cox et al. 1995]. Such variations are referred to as camera distortions.

Intensity variations are also caused by inaccuracies in the modelled position and orientation of the cameras. Both camera distortions and position errors can be reduced through the process of camera calibration [Tsai 1987, Kamberova and Bajcsy 1997]. This is an important component of any real scene reconstruction process and can significantly affect the resulting reconstruction. In this thesis it is assumed that accurate camera calibration has already been performed.

### 3.3.2 Sampling problems

Intensity variations can also arise from the sampling process itself. Although this commonly leads to large variations, it is often overlooked. The main causes of these variations are sampling differences and variations in the imaging convolution kernel between images. These were discussed in more detail in Section 2.3 and Section 2.6.1 and have the most effect in regions of the scene where the surface radiance changes rapidly.

With a discrete scene model the effect of these variations can be reduced by ensuring that the scene sample points are on or near the object surface and that the cameras observe the surface from approximately the same angle. Since the scene is unknown prior to reconstruction, this can be achieved by using a finer sample spacing and only comparing the intensities between cameras with a similar direction of view.

An alternative approach is to interpolate between adjacent samples and then take the best match within half the sample spacing either side of a pixel [Birchfield and Tomasi 1998b, Birchfield and Tomasi 1998a]. This is based on the assumption that the images are adequately sampled, so that no aliasing occurs. Although sometimes untrue, this can easily be enforced by introducing additional focal blur. Sampling variations can also be treated as additional system noise correlated with the intensity differences between adjacent pixels.

One of the problems with the pixel dissimilarity measure presented by Birchfield and Tomasi [1998b] is that it is not extended easily to multiple cameras without using a reference image. This can be avoided by using the symmetric dissimilarity measure suggested by Szeliski and Scharstein [2002]. However, both of these approaches ignore the local intensity gradient and consequently often give false matches. To improve the reliability of matching pixels in a multiple camera system a novel pixel dissimilarity measure is presented in Section 6.4.

### 3.3.3 Transparencies

A common cause of modelling errors resulting in intensity variations is the assumption that the opacities within the scene are fully opaque or transparent. This problem has been investigated in the field of human stereopsis [Weinshall 1993, Weinshall 1991]. Even in the absence of semi-transparent objects such as glass, regions within the scene may appear semi-transparent because of mixed opacities within the region [Baker et al. 1998]. One solution is to allow the scene model to contain semi-transparent regions. This approach has been used in a limited number of reconstruction algorithms and enables the scene to be modelled more accurately [De Bonet and Viola 1999, Szeliski and Golland 1999, Baker et al. 1998]. However, it complicates the model and optimisation process, increasing the number of ambiguities. With most algorithms intensity variations caused by semi-transparent regions are simply treated as outliers.

### 3.3.4 Non-Lambertian surfaces and Radiometric variations

In addition to non-Lambertian surfaces, intensity variations between images can occur due to temporal differences. In many situations the image data are acquired by a single camera which is shifted between images. This movement can affect the lighting of the scene between images. Additionally, unless the environment is static, variations can occur because of external changes. This is particularly important in the field of aerial photography, where there is often a significant delay between images, resulting in changes in lighting and scene due to clouds and relative movement of the sun. As with specular reflections, these errors can be reduced by using a more accurate model or filtering the data to mitigate the effects.

To overcome some of these problems a number of filtering techniques have been developed. These techniques work by reducing the effect of specular reflections or other intensity variations by filtering the data. This is performed either prior to reconstruction or as part of the reconstruction process. A common approach is to compare changes in intensity rather than absolute values, using metrics such as the zero mean normalised cross-correlation or zero mean sum of absolute differences [Aschwanden and Guggenbuhl 1992, Chambon and Crouzil 2004].

Using zero mean metrics is similar to high-pass filtering the data before performing standard correlation or calculating the sum of absolute differences. This approach is based on the assumption that most of the intensity variations are caused by low frequency variations within each image. Although usually true, this assumption is invalid at surface discontinuities or object boundaries. It also has the effect of reducing the signal to noise ratio of the data, since low frequency information about the scene is removed.

Another common approach to reduce the effects of intensity variations is to use colour images, where hue or saturation data are used instead of actual intensities. This is reasonably effective and avoids problems at object boundaries, however, useful intensity information is lost.

Higher level matching primitives can also be used, which are presumed to be insensitive to lighting variations [Tang et al. 2006, Veksler 2002]. These include features such as edges, corners, lines, curves, and textures. Known as feature based matching, this is an extreme case of image filtering, where large amounts of information are discarded in the feature extraction process. This results in a sparse scene reconstruction and non-optimal performance. However, this approach is useful in situations where the images are taken at different times and are subjected to different lighting conditions, such as can occur in aerial or satellite imagery.

Instead of filtering the intensity data, or modelling specular reflections as additional low frequency image noise, specular reflections can be dealt with more accurately by modelling them as an angularly dependent reflectance function. One approach, as proposed by Harding [2001] is to fit a low order polynomial or surface to the set of potentially corresponding pixel intensities rather than a single value. This is done independently for each scene point without any regard to its position or estimated surface normal. Results from a large number of cameras demonstrate the robustness of this technique to specular conditions. However, in most situations, the reconstructed scene is noisier than what would be obtained by fitting a constant to the data. This technique is unsuitable for a small number of cameras, since any scene estimate will tend to fit the data well.

A more complex approach for dealing with specular reflections is to model the intensity variations as a function of surface properties and angle. This is achieved by relating the intensity variation function to the estimated surface normal [Fua and Leclerc 1995].

This complicates the mapping but can help constrain the solution. Additional information and constraints about the expected surface reflectances can also be applied to help further improve the reconstruction.

### 3.3.5 System noise

The final cause of modelling errors is system noise. This is predominately introduced by the cameras and results in random variations in intensity between images. Such variation can be reduced by using high quality camera sensors, however this is expensive. In practice, with good quality cameras and accurate calibration the majority of errors can be attributed to modelling errors.

With most scene reconstruction algorithms the various errors are lumped together and simply treated as additional image noise. Prior statistics about the scene and the distribution of intensity variations can then be used to filter the data or favour the reconstruction of likely scene estimates.

## 3.4 PRIOR INFORMATION

Prior knowledge about a scene can be incorporated into the reconstruction process to improve the estimation process. This allows additional information to be used that is not available from the camera images. In addition to imposing hard constraints on the system model, prior statistical information relating to the likely distribution of scene parameters can be used to guide the reconstruction process. One of the simplest and most commonly used priors is that scene opacities and radiances tend to vary piecewise smoothly throughout the scene. The application of smoothing priors is now considered.

### 3.4.1 Region matching

As mentioned in Section 3.2.1, with traditional stereo matching smoothness priors are usually applied by matching small windowed regions within the image rather than individual pixels. This helps to reduce ambiguities, as larger regions are less likely to be confused with one another than individual pixels. With a volumetric approach this process is equivalent to filtering an initial volume of likelihoods with a rectangular shaped filter in the $x, y$ direction. Based on the assumption that windowed regions are of approximately constant depth, this is effectively low-pass filtering the matching likelihoods before performing the minimisation.

The assumption of constant depth is seldom true and becomes less valid as the window size is increased. The scene within the windowed regions may cover a range of depths, causing the projected view to vary between images. This makes it difficult to compare windows. Consequently, the optimal choice of window varies from

region to region depending on the profile of the surface. To deal with this an adaptive window approach can be used where the window size and shape is modified locally based on the accuracy of constant depth assumption [Kanade and Okutomi 1994, Farid et al. 1994]. Another simple variation is the multiple window approach proposed by Fusiello et al. [1997]. Although this is presented as testing several different windows for each point, it can easily be implemented as a two stage filtering process. Matching likelihoods are first mean filtered, as in standard window based matching, and then maximum filtered before the minimum is selected.

To further improve performance the smoothing function can be extended to 3D by applying a 3D filter. This allows prior knowledge to be implemented more precisely and prevents fronto-planar surfaces from being preferentially reconstructed. However, this approach is not applicable to image based methods, where filtering can only be performed in two dimensions.

Feature-based matching can also be used, where higher level objects called image features are compared. This requires extracting these features from an image and then matching them using some criterion. Typical features include edges, corners, and textures. Because the distribution of features is usually sparse and uneven, the acquired depth map will be incomplete. Extra processing is also required to extract features.

Improved performance can also be achieved through better choice of filtering. Traditionally, smoothness priors have been implemented using mean filters. These are useful for certain types of scenes but are not appropriate around object boundaries where large discontinuities may occur. In such instances median or other such filtering may be more suitable.

Another approach is to iteratively filter scene likelihoods using diffusion or relaxation based techniques [Marr and Poggio 1976, Zitnick and Kanade 1999, De Bonet and Viola 1999, Scharstein and Szeliski 1996, Lee et al. 2001]. These have proved popular in recent years due to their ability to perform complex filtering through a number of relatively simple iterative steps.

### 3.4.2 Segmentation

As with scene opacities, surface radiances are usually correlated between nearby points. This leads to the important observation that discontinuities in depth generally correspond with sharp changes in intensity in each image. This information can be used to improve the scene estimate by favouring the reconstruction of surfaces, whose boundaries correspond with intensity edges in the images. A number of recent algorithms have applied this idea using a segmentation process, where surfaces within the scene are fitted to segments within the images [Lin and Tomasi 2004, Birchfield and Tomasi 1999, Hong and Chen 2004, Bleyer and Gelautz 2007, Sun et al. 2003, Klaus et al. 2006, Yang et al. 2006]. This approach has proved particularly successful, with all of the current

top four algorithms on the Middlebury test set[1] using some form of image segmentation.

## 3.5 OCCLUSIONS

Optimising the objective function for a full system model is extremely difficult because of the complex visibility interaction that occurs between different regions of the scene. As a consequence, a large number of algorithms make various assumptions about the scene visibilities to simplify the system model and optimisation process.

To simplify the problem of dealing with visibilities, three basic approaches can be used. The simplest is to assume that all cameras can see all surface points. Although untrue in most situations, this is a reasonable approximation for scenes containing only a small number of occluded regions. This is the approach used by most traditional depth map based methods, where cameras are usually located close together and face in a similar direction. However, in most cases some occlusions will still occur, often leading to substandard reconstructions. Consequently, this approach is really only suitable for applications such as aerial photogrammetry [Gimel'farb and Zhong 2001, Gruen and Baltsavias 1988, Krupnik 1996], where the scene consists of a single surface visible from all camera positions.

The second approach is to try and estimate a point's visibility by comparing intensities between multiple cameras [Kang et al. 2001]. This can be done in a number of ways. The easiest is to assume that at least $M$ out of $N$ cameras observe each point, with the $M$ cameras chosen to be those most consistent with the data. Alternatively the visibility patterns can be constrained by fitting masks to the set of images [Park and Inoue 1998, Farid et al. 1994]. This allows the spatial relationship between cameras to be used in addition to the observed intensities. A detailed comparison of these techniques is given by Satoh and Ohta [1996].

The third and most accurate approach is to iteratively calculate a point's visibility based on the current scene estimate. The improved visibilities are then used to obtain a new, and hopefully more accurate, estimate of the scene. This process can be performed in a single sweep of the scene volume [Seitz and Dyer 1999] or iteratively over large local search spaces [Kolmogorov and Zabih 2001, Kolmogorov et al. 2003]. Although computationally more intensive, this approach leads to an estimate of the scene that is consistent with the detailed scene model described by Theorem 3. This approach is good for dealing with complex occlusions and is the basis of the dynamic belief propagation algorithm presented in Chapter 6.

[1] See http://vision.middlebury.edu/stereo/

### 3.5.1 Volumetric methods

To more accurately deal with the visibility interaction between scene regions a number of volumetric voxel based methods have recently been proposed [Culbertson et al. 1999, De Bonet and Viola 1999, Kutulakos 2000, Eisert et al. 1999]. These overcome some of the limitations of depth-map models and efficiently utilise a large number of camera images. In these methods a model of the scene is formed directly in 3D scene space so that it best corresponds with the observed images and any prior knowledge. This is more effective than the traditional approach, since information is not lost in projecting what is inherently a 3D estimation problem into 2D.

Volumetric based methods were popularised by Seitz and Dyer [1999] with their 'voxel colouring' algorithm. Here the scene volume is divided into a number of voxels which are traversed in a generalised depth-order. The scene is then reconstructed one layer at a time, moving outward from the cameras. At each depth a voxel's visibility is determined from the already reconstructed nearer voxels. To do this the cameras must be positioned so that voxels can be visited in a near-to-far order relative to every camera. Although efficient, this constraint is a significant limitation as cameras cannot surround or be within the scene.

To allow more general camera positioning, several variations of voxel colouring have been proposed. Kutulakos and Seitz [1998] describe an implementation called 'space carving' where an initially opaque estimate of the scene is progressively carved in several directions until it is deemed consistent. Although this allows arbitrary camera placement, occlusions are not modelled accurately as only the subset of cameras is considered at any stage. A later paper by Kutulakos [2000] describes some additional modifications that enable 'space carving' to compute visibility exactly.

Another approach proposed by Culbertson et al. [1999], called 'Generalised Voxel Colouring' (GVC), computes visibility exactly and allows arbitrary camera placement. In this case the consistency of all opaque surface voxels is considered at each iteration and, similar to before, voxels are carved if they are inconsistent. Eisert et al. [1999] also present a variation of GVC called 'multi-hypothesis voxel colouring'. With this approach a set of hypotheses is first identified for each voxel. These are then narrowed down during a hypothesis removal step until only consistent voxels remain.

Although these approaches allow more general camera placements, voxel consistency is still used locally to determine whether or not a voxel is opaque. Because of intensity variations between the images, the MSE of the back-projected data is compared with a threshold to determine whether a voxel is opaque or not. Using such a threshold causes problems as an adequate rather than optimal solution is obtained. If too high a threshold is used, the carving process stops before the true minimum is reached. If too low a threshold is used, many valid but noisy voxels will be carved away resulting in an incomplete scene.

To overcome these problems, Slabaugh et al. [2000b] proposed a volumetric optimisation method that refines a reconstruction to minimise reprojection error. This begins with a scene estimate obtained through one of the voxel colouring methods and then attempts to reduce an error measure by adding or removing voxels from the estimate. The resulting scene is more likely and improves the reconstruction as demonstrated. However, because only small moves are allowed, the solution is prone to remaining in a local optimum near the initial conditions. This results in local rather than global optimisation. A detailed survey of volumetric methods is given in Slabaugh et al. [2001]. A more recent survey is given in Seitz et al. [2006]. However, the problem with most of these approaches is that the global optimisation techniques used are poor, resulting in sub-optimal scene reconstructions.

Silhouettes of the scene object or group of objects, obtained by segmenting out the background, can also be used to directly reconstruct a model of the scene, or obtain visibility information about regions within the scene [Vogiatzis et al. 2005, Tran and Davis 2006]. This approach is particularly suited to controlled environments, where the background radiance is uniform and a large number of cameras are used.

### 3.5.2 Tomographic approach

One of the first voxel based approaches to the scene reconstruction problem was presented by Preddey and Lane [1997]. In this work the observed camera images are back-projected onto a 3D reconstruction grid. The back-projected data are then used to form an estimate of the visible surfaces using one of two approaches. The term tomographic is used here to indicate the reconstruction of a volume from a number of projections and does not imply that the actual interior of objects is reconstructed.

The first approach is a variation of the CLEAN algorithm [Hogbom 1974], developed for de-blurring astronomical images when the Point Spread Function (PSF) is known. Using this approach the back-projected images are summed at each scene point, forming a set of values that are equivalent to the visible surface blurred by a spatially variant PSF. The CLEAN algorithm is then applied in an attempt to de-blur the scene and reconstruct the visible surfaces. Unfortunately, since the PSF is generally unknown at each scene point some approximations must be made. With their CLEAN approach Preddey and Lane [1997] assume that all surfaces are visible to all cameras, resulting in a constant PSF that is simply a function of the camera positions. As demonstrated in their work, this approach can produce approximate results for simple scenes. However, the reconstructions are relatively noisy and computationally intensive.

The second method presented by Preddey and Lane [1997] for estimating visible surfaces from the back-projected data uses a probabilistic approach where a probability is assigned to each scene point based on how likely it is on a visible surface. A maximum likelihood approach is used, where the prior probability of a point being on a visible

surface, as well as the probability of obtaining the data, is assumed constant. The most likely point along each line of sight from the central camera is then assigned as a surface point. Following this, a threshold is applied to eliminate any poorly distinguished points. The result is a partial reconstruction containing mainly those surfaces that are visible to all cameras. Partially occluded surfaces can then be reconstructed by removing all image data corresponding to the reconstructed surfaces and repeating the above process. The procedure may be repeated further for multiple layers of occlusion.

This second approach yields good results when a large number of cameras are used and is also computationally efficient. However, it makes no use of any prior information other than the visibility constraint, and as a result the obtained reconstructions are reasonably noisy when only a few cameras are used. Following on from the work of Preddey and Lane [1997], Harding et al. [2000] apply the volumetric approach to the problem of robot navigation.

### 3.5.3 Gimel'farb's method

Another early voxel based approach was proposed by Gimel'farb and Haralick [1997] in their multiple camera approach to scene reconstruction. This work was aimed at reconstructing digital elevation maps (DEMs) from aerial photographs. To reduce computational requirements the voxel space was represented as a depth map, restricting the scene to a single, but possibly discontinuous, surface. Although suitable for their application, this simplification takes away many of the advantages of a truly 3D voxel representation.

The algorithm presented by Gimel'farb and Haralick [1997] consists of two main stages. The first stage involves calculating a dissimilarity measure for each voxel, based on the observed image data, while the second stage is a refinement stage, dealing with occlusions and regularising the reconstruction. To begin, a set of grey level values, corresponding to the projected intensities of a point, are associated with each voxel. A dissimilarity measure $v$ is then calculated for each voxel based on these values. This is given by

$$v = (\max\{0, \varepsilon_{\min} g_{\max} - \varepsilon_{\max} g_{\min}\})^2, \tag{3.2}$$

where $g_{\max}$ and $g_{\min}$ are the maximum and minimum observed intensities respectively, and $\varepsilon_{\min}$ and $\varepsilon_{\max}$ are numbers which bound the admissible variations in observed intensity of a given point. Having calculated a set of dissimilarities, the height $Z$ giving the smallest value of dissimilarity for each $X, Y$ location is chosen as an initial estimate of the surface at that position; see Fig. 3.3(a). If the dissimilarity measure is the same for several heights, then the point is likely to lie in a homogenous region. In this case the smaller height is chosen. This is somewhat arbitrary but seems to give better results as homogenous regions appear to be more likely in the background. The range in intensities is used as a confidence measure for each $X, Y$ position, with larger ranges being less

confident.

In the second stage of the reconstruction process, the scene estimate is refined by checking for possible occlusions. To begin with a confidence measure, $r = g_{\max} - g_{\min}$, is assigned to each voxel. If a less confident voxel occludes a more confident one from any camera position, then that voxel is removed, reducing the surface height at that $X, Y$ position; see Fig. 3.3(b). A new confidence measure is then assigned to that point, corresponding to the range of intensities at the new height. This process is repeated until no further changes occur. The effect is to help remove any false spikes that may occur in the first estimate.

Following the removal of possible false occlusions, the scene estimate is further refined by median filtering the resulting depth map. Rather than simply taking the median over some moving window, the median is only taken of those points that are more or equally confident than the central point within the window, and form a continuous surface around it assuming 4 connectivity; see Fig. 3.3(c). This helps remove noise from the estimate by favouring continuity within surfaces that have a high confidence.

To demonstrate the approach of Gimel'farb and Haralick [1997], an implementation of the algorithm was applied to the 'head scene', shown in Fig. 3.4, courtesy of the University of Tsukuba. The values $\varepsilon_{\min}$ and $\varepsilon_{\max}$ were both set to one, as this produced the best results in this situation. The ideal depth-map and obtained results using two cameras and a range of 16 depths are shown in Fig. 3.5.

## 3.6 OPTIMISATION TECHNIQUES

There have been many approaches to optimise or calculate the mean of the joint probability distribution in the scene reconstruction problem [Scharstein and Szeliski 2002, Szeliski et al. 2006, Gargallo and Sturm 2005, Dempster et al. 1977, Strecha et al. 2004, Kolmogorov and Rother 2006, Meltzer et al. 2005, Gargallo and Sturm 2005].

### 3.6.1 Sampling techniques

One of the simplest and most general approaches to finding the maximum or mean of a distribution is to sample the distribution. This can be done in a number of different ways. A common approach is to simply sample the distribution over a uniform grid of points and take the maximum or mean of the samples as the optimal solution. If the distribution function is reasonably smooth, then the true minimum will generally be within half the grid spacing in each direction. This approach works well for functions with few variables. However as the number of variables increases, the solution space increases exponentially and the number of grid points becomes prohibitively large. In addition, if the function is not sampled adequately then the true minimum may be missed altogether. Despite these disadvantages, the grid search approach is still effective

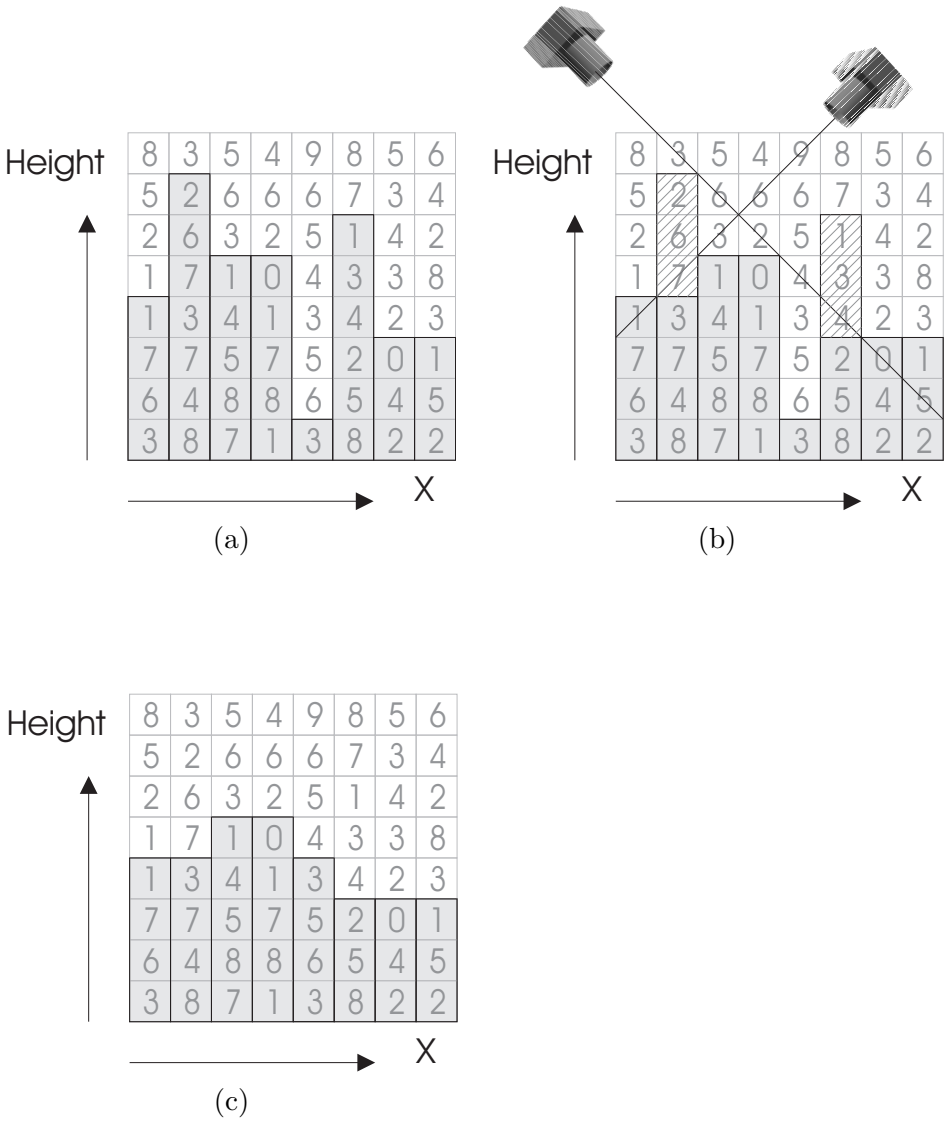


**Figure 3.3** Steps in terrain reconstruction algorithm [Gimel'farb and Haralick 1997]. (a) An initial estimate of the scene is formed by selecting, at each $X, Y$ position, the height giving the smallest value of dissimilarity. (b) The scene estimate is then refined by removing all voxels which occlude a more confident one. (c) Finally, median filtering is applied to the resulting depth map to help remove spurious values.

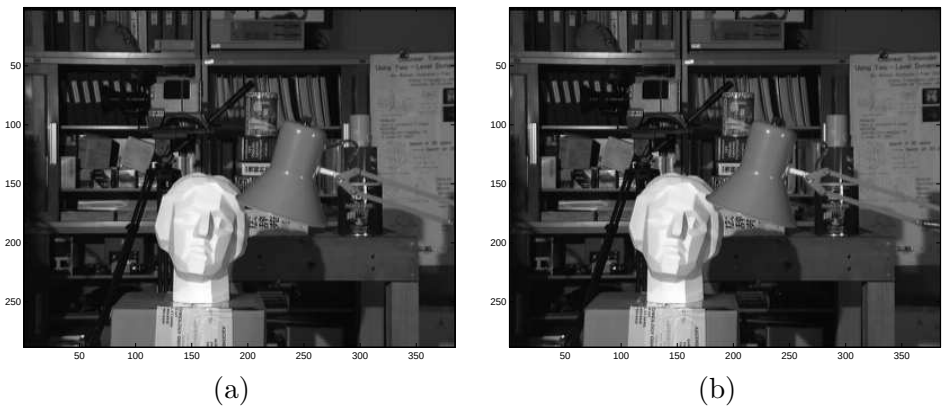


**Figure 3.4** Head scene. (a) Left and (b) right images.

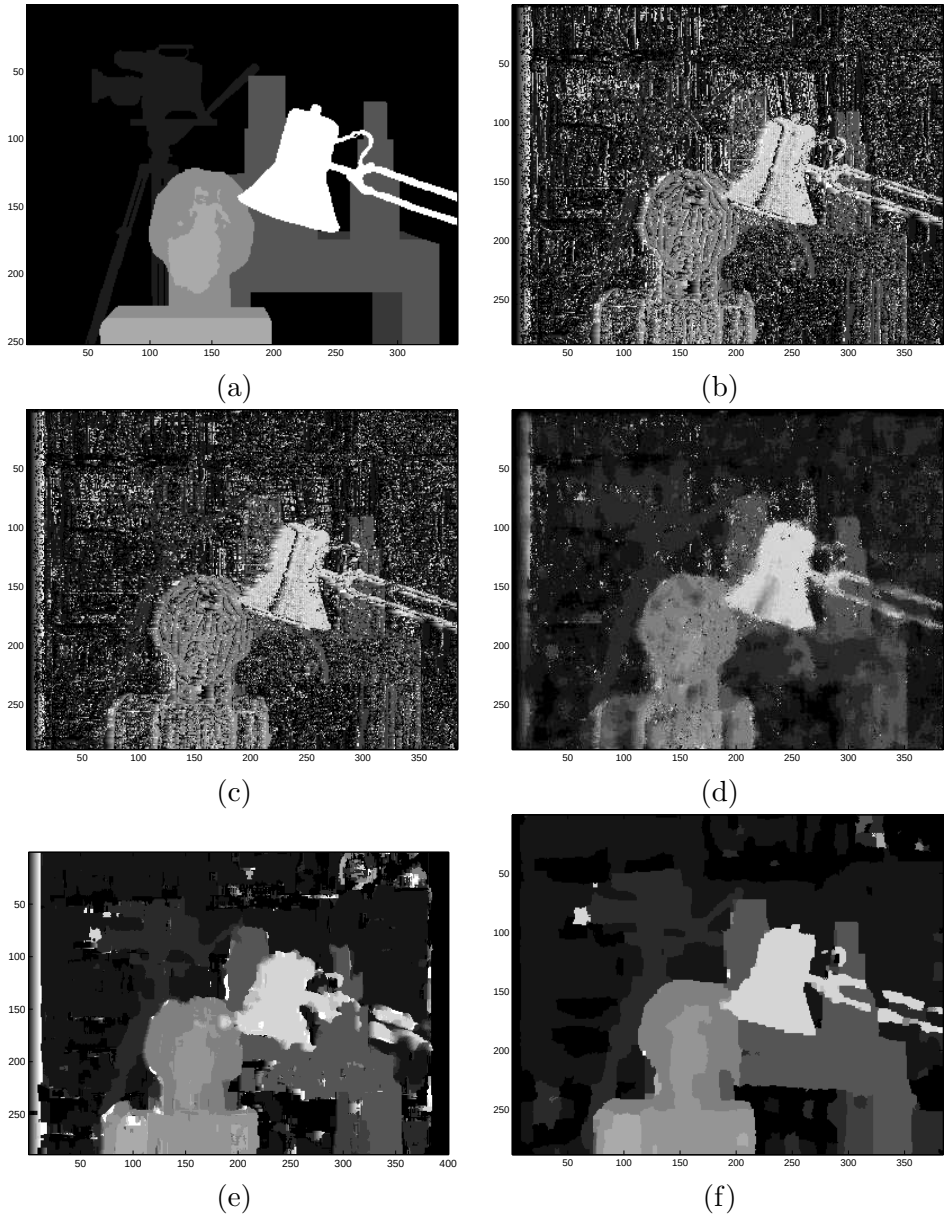


**Figure 3.5** Results from terrain reconstruction algorithm [Gimel'farb and Haralick 1997]. (a) Ideal depth map (b) Initial depth map, obtained by selecting at each $X, Y$ position, the height giving the smallest value of dissimilarity. (c) The refined depth map after occlusions have been dealt with. (d) Final depth map, obtained by median filtering. (e) Depth map produced by implementation of a simple area based correlation algorithm with a window size of 9x9. (f) Depth map produced by implementation of Sun et al. [2002] belief propagation algorithm.

in certain situations and is the basis of traditional stereo matching algorithms, where the matching objective is minimised locally as a discrete function of depth.

Instead of sampling the joint probability distribution at fixed number of points, a sequence of samples can be generated either randomly or deterministically. One of the most commonly used approaches is the Metropolis-Hasting algorithm [Metropolis et al. 1953, Hastings 1970]. This is a rejection sampling algorithm which generates a random sequence of samples forming a Markov chain. The advantage of this approach is that most samples are from regions where the posterior probability is high. A special case of this is Gibbs sampling [Geman and Geman 1984]. The biggest problem with these sampling algorithms for stereo reconstruction, is that a very large number of samples are required to obtain a good scene estimate, making this approach very slow.

### 3.6.2 Continuous optimisation

Numerous techniques for optimising continuous functions have been proposed, the success of which depends on the objective function and quality of the initial estimate. A number of these approaches have been applied to the scene reconstruction problem with reasonable success [Faugeras and Keriven 1997, Faugeras and Keriven 1998].

An alternative approach is to use discrete methods to optimise a continuous functions. This approach has been recently applied to the scene reconstruction problem by Paris et al. [2006], where graph cuts are used to minimise a continuous function up to a discretisation.

### 3.6.3 Local methods

Another commonly used class of minimisation techniques are gradient methods, where an initial estimate is progressively refined by descending downhill until a minimum is reached. These are local techniques, therefore requiring a good initial estimate of the scene if reasonable results are to be obtained. Commonly used to solve sparse systems of linear equations, these techniques are defined by their descent direction and step size. The simplest approach is to step in the direction of maximum gradient. Known as steepest descent, this approach usually sets the step size to correspond with the minimum along the search direction. For linear systems or quadratic functions this is easily found directly from the gradient.

With non-linear functions various line search techniques must be used. These are usually iterative techniques such as the Newton-Raphson and secant methods, based on approximating the function with a quadratic. Having determined the step size, a new estimate can be obtained and used as the starting point for the next iteration. This process is repeated until the solution converges or a certain number of iterations have been performed. Rather than stepping in the direction of steepest descent, other

techniques can be used, such as conjugate gradient methods. These typically converge more rapidly and are therefore often chosen in preference to steepest descent algorithms.

'Linear least squares' is often used for continuous depth map models. Prior knowledge can be applied directly to the problem and an overall minimum can be found using least squares techniques. With this approach the error function is linearised around some operating point and then solved using linear methods. This is another iterative technique, where the solution to the linearised problem is used as the operating point for the next iteration. The process is usually repeated until convergence is reached or a certain number of iterations have been performed. Especially popular in the photogrammetry field, this technique has been used by a number of researchers for reconstructing digital elevation maps (DEMs) from aerial photographs [Gimel'farb and Zhong 2001, Gruen and Baltsavias 1988, Rosenholm 1987, Gruen and Baltsavias 1987, Krupnik 1996].

In situations where least squares are applied a reasonable estimate of the scene is usually already known and the objective function is fairly pliant around the minimum point. However for more general scenes, where the objective function is highly non-linear and a good initial estimate is unavailable, the approach is rather limited and the solution will tend to get stuck in a local minimum if it converges at all.

### 3.6.4 Iterative refinement

Iterative refinement refers to a broad class of optimisations techniques. Iterative refinement algorithms progressively improve the solution through a number of local changes. Starting with an initial configuration, a sequence of configurations are generated within the neighbourhood of the current configuration. If any of these improves the objective function, the new configuration is accepted and the process is repeated. This technique has recently been used by Slabaugh et al. [2000b] with reasonable success to improve the performance of voxel colouring algorithms.

### 3.6.5 Graph cuts

Over the last few years a number of novel reconstruction algorithms have been developed based on graph cuts [Lin and Tomasi 2004, Tsin et al. 2003, Kim et al. 2003, Snow et al. 2000, Kolmogorov and Zabih 2004, Freedman and Drineas 2005, Kolmogorov et al. 2003, Kolmogorov and Zabih 2002, Kolmogorov and Zabih 2001, Tran and Davis 2006, Vogiatzis et al. 2005, Hong and Chen 2004, Bleyer and Gelautz 2007, Ishikawa and Geiger 1998, Ishikawa 2003]. These attempt to minimise an objective function by formulating it as a labelling problem, solved by finding the minimum multi-way cut through a corresponding graph. In general, the multi-way graph cut problem is NP hard and must be solved using approximate techniques. This usually involves breaking the problem into a number of two terminal subproblems, for which efficient and optimal

solutions exist. Results from these techniques are promising, outperforming most other algorithms in 2005[2].

One of the earliest uses of graph cuts was the maximum-flow formulation by Roy and Cox [1998]. The scene volume was divided into a number of interconnected nodes, with the nearest nodes connected to a source terminal and the farthest nodes connected to a sink terminal. The minimum cut through this graph then corresponded directly to an object's surface. Although reasonable results were obtained, this approach did not take account of visibilities and made poor use of prior knowledge, reconstructing only those points which were visible within some reference image.

Another use of graph cuts has been to determine voxel occupancy from a number of foreground and background images [Snow et al. 2000]. In this situation the scene volume is represented as a grid of interconnecting nodes, each of which is connected to both a source (transparent) and sink (opaque). The minimum cut then partitions the nodes into these two sets. Again this approach does not make full use of the camera data, nor model visibilities.

More recently Kolmogorov et al. [2003] have published a multiple camera graph cut algorithm which explicitly models occlusions and treats all cameras symmetrically. In their formulation, an objective or energy function consisting of three terms is iteratively minimised by finding the local minimum within one $\alpha$ expansion of the current estimate. This involves converting the energy minimisation to a binary labelling problem at each stage, that is then solved by finding the minimum cut through an appropriate graph. In this case the reconstructed graphs are simply used as a means of minimising the local energy function and have no direct correspondence to the final scene estimate.

### 3.6.6 Dynamic programming

Another approach to the reconstruction problem is to use dynamic programming techniques to find the minimum path through a set of likelihoods [Sun 1999, Cox et al. 1996, Meerbergen et al. 2002, Intille and Bobick 1994, Ohta and Kanade 1985, Gimel'farb 1998]. First invented in 1953 by Bellman [1960], dynamic programming is a common technique, widely used to solve a number of discrete optimisation problems. This is done by breaking a multi variable problem into a number of single variable problems solved sequentially. To do this, the underlying process must be Markovian, where the optimal decision at each stage depends only on the current state and not on how this state was reached. In the case of scene reconstruction, the easiest way to do this is to have stages corresponding to points along some line in a reference image and letting a point's depth depend only on the depth at the previous stage and not on the depth of any other stages. The dynamic programming approach produces the most likely path through this network, giving the depth of all points along the reference line. This process can

[2]See Middlebury stereo evaluation version 1 - http://vision.middlebury.edu/stereo/

be repeated for various lines in the reference image allowing a full reconstruction of the depth-map to be obtained.

The dynamic programming technique has a number of drawbacks. Firstly, only those surfaces visible within the reference image are reconstructed. This leads to a number of problems, since the interaction between scene points cannot be fully accounted for. Because of this, both prior knowledge and scene visibilities are not modelled properly. Secondly, as the likelihood of a point's depth may depend only on the depth of its previous neighbour, many types of prior knowledge cannot be applied. This also applies to scene visibilities, which depend on the whole scene (not just neighbouring points) leading to errors when evaluating the state likelihoods.

To help overcome some of these limitations Ohta and Kanade [1985] propose an inter and intra scanline approach where dynamic programming is applied in both the horizontal and vertical direction. Another variation is the maximum surface technique proposed by Sun [1999]. These improve the application of prior knowledge by allowing a point's likelihood to be influenced from both neighbouring directions. Nevertheless, most of the existing problems still remain.

### 3.6.7 Consistency thresholding

Another recent approach is to use a consistency threshold as a means to determine a point or region's opacity. This is the basis of voxel colouring [Seitz and Dyer 1999] and space carving algorithms [Kutulakos and Seitz 1998], where points are chosen as opaque if they are consistent with the camera images. To determine consistency, the observed intensities of a point are compared, giving a measure of similarity compared to some pre-defined threshold. This threshold is usually chosen based on the system noise, estimated prior to reconstruction. The problem with this approach is that decisions are made locally, resulting in a non-optimal overall estimate of the scene. This often results in an enlarged reconstruction or produces gaps in the estimated scene [Slabaugh et al. 2000b].

### 3.6.8 Stochastic algorithms

A problem with iterative improvement algorithms is that they terminate in the first local minimum, defined by the search neighbourhood. If this search neighbourhood is large, the local minimum will be strong, producing better results. However, the same problem still remains. To help overcome this, several stochastic algorithms have been proposed, where non-improving steps are probabilistically accepted. This enables the algorithm to escape local minima and converge on the global minimum.

Perhaps the most common example is simulated annealing. This has been applied to the stereo problem by Ouali et al. [1996]. Based on the physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure, this process has

been used to solve a large number of combinatorial optimisation problems. To ensure convergence, the probability of a non-improving step is slowly decreased over time until it is zero. This is referred to as the cooling schedule. If performed infinitely slowly, the algorithm is guaranteed to converge to the global minimum. However, with finite cooling schedules this guarantee is lost. The problem with simulated annealing is that convergence speed is very slow, making it impractical for many applications. In fact Greig et al. [1989] demonstrate that practical implementations of simulated annealing give results that are very far from optimum even in the relatively simple case of binary labelling.

### 3.6.9 Genetic programming

An alternative approach to the reconstruction problem is to use genetic algorithms to search for an optimal solution. Based on biological evolution, these algorithms create a population of solutions which evolve over time. To begin, an initial population of solutions is formed. Individual solutions within the population then fight for survival based on how well they work. The best solutions are combined and mutated giving rise to the next generation of solutions. The process is then repeated until some stopping criterion is reached.

Although genetic programming is currently an active area of research, genetic algorithms have only seen limited use in scene reconstruction. Perhaps the most promising work is that by Gong and Yang [2001], who use genetic programming to minimise a disparity map energy function. In their approach the scene is represented as a disparity image using a quad-tree data structure. The genetic algorithm is then used to search for the representation with minimum energy. Although reasonable results are obtained, the disparity map representation means that occlusions and prior knowledge cannot be accurately modelled.

### 3.6.10 Greedy algorithms

Another popular technique, which forms the basis of many optimisation algorithms, is to iteratively refine a solution through a sequence of locally optimal steps. Commonly referred to as a greedy approach, this technique accepts the best local solution at each stage, with the aim that this will lead to a global optimum. Strictly speaking the greedy approach only applies to problems where the solution is progressively constructed by adding elements from an initial solution set. In this case the algorithm begins with an empty set and sequentially adds those elements which most improve the current objective.

The term greedy algorithm is also used more widely to describe any algorithm which performs a sequence of locally optimal steps in order to find a global optimum. The advantage of this approach over other search techniques is only a single path must be

considered, making it comparatively fast. Generally, it is also straightforward to implement and easy to understand. However, one problem is that locally optimal improvements will not necessarily yield an optimal global solution. Therefore, it is important to formulate the local solution spaces appropriately. One technique is to form a strong local minimum at each stage, by making the local solution space as large as possible. Another approach is to start with a good initial estimate, as this reduces the search space, decreasing the chances of finding a false minimum.

Region growing is another technique used to progressively grow surfaces until a complete estimate of the scene is formed. This is done by selecting a set of seed points and then adding adjacent points to the partial reconstruction until the estimate is complete. This is usually achieved through a greedy approach where the most likely adjacent points are added first. To work effectively, every visible surface within the scene must contain at least one seed point, and furthermore, every seed point should correspond with some surface. For general scenes this is rather difficult to achieve unless the scene is known, in which case the problem is already solved. However, a reasonable reconstruction can be achieved if a number of seed points can be found on the major surfaces.

Chen and Medioni [1999] used a variant of the region growing technique where the scene is grown simultaneously from all current surface points. In their work when the projection of two surface fronts meet, the most likely surface erodes away the other. Initial seed points are chosen to be unique maxima in the direction of the reference camera and above a given threshold.

### 3.6.11 Diffusion algorithms

Diffusion algorithms are a further class of minimisation techniques especially popular for dealing with complex or stochastic systems. These are iterative techniques where local information is propagated throughout the system in an attempt to find a global minimum. Especially well suited to parallel implementation, these techniques are commonly believed to be the basis of biological stereopsis [Marr and Poggio 1976]. Marr and Poggio [1976] demonstrated this idea with their cooperative algorithm for modelling biological vision systems. Using uniqueness and continuity constraints, prior knowledge was applied through a cooperative process, helping to remove matching ambiguities.

Originally designed for matching binary features between two images, the cooperative approach was later extended by Zitnick and Kanade [1999] to work on greyscale images. In the intervening and subsequent years, many examples of the cooperative approach have been developed, mainly in the fields of human stereopsis and visual perception [Mansson 1998, Henkel 1997]. A variation, based on Bayesian estimation, is the non-linear diffusion algorithm presented by Scharstein and Szeliski [1996].

### 3.6.12 Belief propagation

Belief propagation is another type of global optimisation algorithm, where the likelihood of each scene variable is iteratively updated based on local message passing. These messages represent the probability that the receiver should be in a particular state based on all the current information from neighbouring variables. This technique is proving to be increasingly popular in the solution of many estimation problems, and has recently been applied to the reconstruction problem with considerable success [Sun et al. 2003, Sun et al. 2005, Klaus et al. 2006, Yang et al. 2006, Zitnick and Kang 2007, Larsen et al. 2006, Guan and Klette 2008]. Currently all of the top 5 stereo reconstruction algorithms on the Middlebury test set[3] use belief propagation. A detailed description of the belief propagation algorithm is given in Chapter 5.

The use of belief propagation for stereo reconstruction was first demonstrated by Sun et al. [2002]. With their approach the scene is represented as a depth map relative to one of the images. The conditional likelihood of each point given the image data is then modelled using an additional set of observation nodes and associated compatibility functions. One of the main problems with their method is that the visibility interaction between scene points is poorly modelled. In particular they assume occlusions are statistically independent of the estimated depth map.

To improve the scene reconstruction, a variety of different approached based on belief propagation have been proposed to account for the visibility interactions between scene parameters [Forne and Hayes 2002, Forne and Hayes 2003, Sun et al. 2005, Yang et al. 2006, Zitnick and Kang 2007, Larsen et al. 2006].

A recent variation of the max-product belief propagation algorithm is the Tree Re-Weighted (TRW) max-product message passing algorithm presented by Wainwright et al. [2005]. A provably convergent variation is given by Kolmogorov [2006]. As demonstrated by [Szeliski et al. 2006, Kolmogorov and Rother 2006, Meltzer et al. 2005], TRW gives improved results on several standard stereo systems compared with standard belief propagation. TRW also has been demonstrated to be optimal for one particular stereo model [Meltzer et al. 2005].

For continuous systems, an alternative message passing algorithm referred to as expectation propagation can be applied [Minka 2001a, Minka 2001b]. This is a generalisation of belief propagation that allows continuous distributions to be optimised. This could be useful for obtaining high accuracy depth-maps without having to use a large number of discrete states.

[3] See http://vision.middlebury.edu/stereo/

# Chapter 4

## GREEDY ALGORITHM

In the previous chapter a variety of techniques for reconstructing three dimensional scenes from multiple camera images were discussed. One of the biggest problems facing these algorithms is that the likelihood of a scene point being opaque depends strongly on its visibility. These visibilities are determined by the state of numerous points within the scene, resulting in a complex joint probability distribution. This interaction between scene points complicates the optimisation process, since the resulting reconstruction function is difficult to optimise. To simplify the problem, most algorithms make a number of assumptions about scene visibilities. Unfortunately in many cases these assumptions are incorrect and are often inconsistent with the resulting scene estimate.

To help deal with the visibility interaction between points and improve the scene estimation process an efficient novel voxel based algorithm for finding an approximate Maximum A Posteriori (MAP) estimate of the scene is presented in the chapter. The proposed algorithm progressively reconstructs the scene while updating the scene visibilities, so that the estimation of each voxel is consistent with the overall scene estimate. Beginning with a transparent volume, voxels are progressively assigned as opaque until a complete scene estimate has been formed. This is done by selecting the most likely surface voxel at each iteration and updating the remaining visibilities and associated probabilities accordingly.

The problem of finding the MAP scene estimate is expressed as a minimisation problem in Section 4.1. Assuming the visibility of all surface points is known and that the prior likelihoods of all scenes are equal, this can be expressed as a pixel ray assignment problem. An efficient algorithm for solving this is presented in Section 4.3. In Section 4.4, visibility interactions between voxels is introduced. To improve results, an alternative greedy algorithm is presented in Section 4.5. Basic smoothness priors are introduced in Section 4.6. Several improved algorithms for incorporating prior information are described in Chapter 5 and Chapter 6. An efficient iterative maximisation technique is presented in Section 4.7, followed by a brief discussion of the results in Section 4.8.

## 4.1 MAP ESTIMATE

By formulating scene reconstruction as a Maximum A Posteriori (MAP) estimation problem, and using a voxel based scene representation, the reconstruction problem can be expressed as an optimisation problem where the objective is to assign an opacity and radiance to each voxel, so that the joint posterior probability distribution over these parameters is maximised. This can alternatively be expressed as a minimisation problem, where the objective is to minimise a weighted combination of the projected error and negative log prior probability of the scene.

Using $\mathbf{S} = \{S_1, S_2, \ldots, S_M\}$ to represent the set of scene parameters and $\mathbf{C} = \{C_1, C_2, \ldots, C_N\}$ to represent the set of camera pixel intensities, the MAP reconstruction problem can be expressed as given $\mathbf{C}$ equals $\mathbf{c}$, find the most likely estimate $\mathbf{s}$ of $\mathbf{S}$. This can be written using Bayes' rule as

$$S_{\text{MAP}}(\mathbf{c}) = \arg\max_{\mathbf{s}} \left[ \frac{\rho_{C|S}(\mathbf{c}|\mathbf{s})\rho_S(\mathbf{s})}{\rho_C(\mathbf{c})} \right]. \tag{4.1}$$

The denominator term, $\rho_C(\mathbf{c})$, represents the prior probability of obtaining the observed camera data. Since this term is independent of $\mathbf{s}$, it can be removed from the expression without affecting the optimisation. The first numerator term $\rho_{C|S}(\mathbf{c}|\mathbf{s})$, represents the likelihood of observing the data $\mathbf{c}$ given estimate $\mathbf{s}$. Using $\check{\mathbf{c}}$, to represent the pixel intensities that would be recorded in the absence of any noise, abberations, or modelling errors, this can be equivalently written as

$$\rho_{C|S}(\mathbf{c}|\mathbf{s}) = \int_{\check{\mathbf{c}}} \rho_{C|\check{C},S}(\mathbf{c}|\check{\mathbf{c}}, \mathbf{s})\rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s}), \tag{4.2}$$

where $\rho_{C|\check{C},S}(\mathbf{c}|\check{\mathbf{c}}, \mathbf{s})$ is the probability distribution of obtaining $\mathbf{c}$, given $\check{\mathbf{c}}$ and $\mathbf{s}$, and $\rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s})$ is the probability distribution of $\check{\mathbf{c}}$, given $\mathbf{s}$. Assuming independent noise at each of the sensors, this can be simplified to give

$$\rho_{C|S}(\mathbf{c}|\mathbf{s}) = \int_{\check{\mathbf{c}}} \rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s}) \prod_{k=1}^{N} \rho_{C_k|\check{C}_k}(c_k|\check{c}_k). \tag{4.3}$$

In most instances, the ideal intensity at each pixel will be uniquely determined by the scene parameters, and can be expressed as $\check{c}_k = \text{proj}_k(\mathbf{s})$, where $\text{proj}_k(\mathbf{s})$ is the projection of $\mathbf{s}$ onto the $k^{\text{th}}$ pixel. For infinite scenes, or any semi-infinite scenes that include all points that are within the field of view of the cameras, this is guaranteed. It is also true for finite scenes, provided that there are no radiating or opaque regions outside the scene that are visible in any of the cameras. In situations where points outside the defined scene volume are visible, the probability distribution $\rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s})$ can be simplified by assuming that the ideal pixel intensities depend on the radiance of regions inside or outside the scene but not both. This condition will be valid provided

that radiances outside the modelled scene volume are independent of those within the scene, there are no opaque or radiating surfaces between the defined scene volume and any of the cameras, and that the scene is either completely transparent or opaque along any pixel beam.

Assuming binary transmittances through the scene along each pixel beam, and using $\xi_k(\mathbf{s})$ to represent a boolean function that is equal to one if the transmittance along the $k^{\text{th}}$ pixel beam is zero, and zero otherwise, the conditional probability distribution $\rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s})$ can be expanded to give

$$\begin{aligned}\rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s}) &= \rho_{\xi_1(\mathbf{s})}(\check{\mathbf{c}}_{\xi_1}(\mathbf{s})|\mathbf{s})\rho_{\xi_0(\mathbf{s})}(\check{\mathbf{c}}_{\xi_0}(\mathbf{s})|\mathbf{s}) \\ &= \prod_{k\in\check{C}_{\xi_1}(\mathbf{s})} \delta(\check{c}_k - \text{proj}_k(\mathbf{s})) \times \rho_{\xi_0(\mathbf{s})}(\check{\mathbf{c}}_{\xi_0}(\mathbf{s})|\mathbf{s}),\end{aligned} \tag{4.4}$$

where $\check{C}_{\xi_1}(\mathbf{s})$ is the set of pixels for which $\xi_k(\mathbf{s}) = 1$ and $\check{C}_{\xi_0}(\mathbf{s})$ is the remaining set of pixels, corresponding with $\xi_k(\mathbf{s}) = 0$. For pixel rays outside the scene, $\xi_k(\mathbf{s}) = 0$. The term $\rho_{\xi_0(\mathbf{s})}(\check{\mathbf{c}}_{\xi_0}(\mathbf{s})|\mathbf{s})$ represents the joint probability distribution of obtaining the ideal pixel intensities in the set $\check{C}_{\xi_0}(\mathbf{s})$. This function is governed by the prior probability of the background radiances. In situations where the background radiance is known, this term will be a delta function. If the background radiances are unknown, a uniform distribution over the range of pixel intensities is usually assumed, allowing the term to be approximated by $1/\kappa^n$, where $n$ is the number of pixels in $\check{C}_{\xi_0}(\mathbf{s})$, and $\kappa$ is the dynamic range of the cameras.

This function can be further simplified by assuming the average transmittance through the scene along any pixel beam is zero. Such a scene is referred to as *complete*, as it completely defines all ideal pixel intensities [Seitz and Dyer 1999]. For scenes with binary regional opacities, this condition is ensured if there is at least one opaque region extending across every pixel ray. With infinite or semi-infinite scenes, an equivalent scene can always be found that is complete with respect to the set of camera images. This is achieved by replacing any transparent region extending to infinity along incomplete pixel rays, with an opaque region. So long as the transmitted radiance of the two regions is the same, both will appear identical from all camera positions.

By ensuring the scene estimate is complete, the conditional probability distribution $\rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s})$, can be simplified to give

$$\rho_{\check{C}|S}(\check{\mathbf{c}}|\mathbf{s}) = \prod_{k=1}^{N} \delta(\check{c}_k - \text{proj}_k(\mathbf{s})). \tag{4.5}$$

Substituting this back into Eq. 4.3, gives

$$\rho_{C|S}(\mathbf{c}|\mathbf{s}) = \prod_{k=1}^{N} \rho_{C_k|\check{C}_k}(c_k|\,\text{proj}_k(\mathbf{s})). \tag{4.6}$$

Assuming binary regional transmittances, the projection $\text{proj}_k(\mathbf{s})$ of $\mathbf{s}$ onto the $k^{\text{th}}$ pixel is given from Theorem 4 in Chapter 2 as

$$\text{proj}_k(\mathbf{s}) = \overline{R}_{\nu i}(x_k, y_k, Z_i^*(x_k, y_k, \overline{T}_\nu)), \tag{4.7}$$

where $i$ is the index of the image containing pixel $k$, $Z_i^*(x_k, y_k, \overline{T}_\nu)$ is the depth of the nearest opaque region along the $k^{\text{th}}$ pixel ray, and $x_k$ and $y_k$ are the $x$ and $y$ coordinates of the $k^{\text{th}}$ pixel.

Using a voxel based scene model where each voxel is represented by its radiance, $r_j(\theta)$, and binary opacity, $\alpha_j$, the regional transmittance $\overline{T}_\nu$, and transmitted radiances, $\overline{R}_{\nu i}$, for each camera, are found by filtering and interpolating between the opacity and transmitted radiances of surrounding scene voxels. This complicates the inverse mapping as the radiance of numerous voxels will affect the observed intensity of each pixel. To simplify the optimisation, the joint conditional probability function, $\rho_{C|S}(\mathbf{c}|\mathbf{s})$, can alternatively be re-expressed so that the filtering and interpolation are performed in the image domain, rather than in scene space.

By modelling the observed pixel intensities in each image as sample points of a continuous intensity distribution, the joint conditional probability function, $\rho_{C|S}(\mathbf{c}|\mathbf{s})$, can be closely approximated as a product of local conditional probability distributions over the set of perturbed pixel intensities, $\acute{\mathbf{C}} = \{\acute{C}_1, \acute{C}_2, \ldots, \acute{C}_N\}$, where $\acute{C}_k$ lies within half a pixel width of $C_k$. The set of pixel rays corresponding with possible perturbed positions of a given pixel $C_k$, define a rectangular cone in space that will be referred to as the extended pixel ray of pixel $C_k$. If the position of $\acute{C}_k$, denoted $x_{\acute{k}}$, is chosen to correspond with the projected image position of the nearest opaque voxel along the $k^{\text{th}}$ extended pixel ray and the discrete depths, defined by $Z_{\acute{k}}^*(x_{\acute{k}}, y_{\acute{k}}, \overline{T}_\nu)$, coincide with the voxel depths, then the imaging sample points will coincide with the voxel positions. This avoids the need for interpolating between scene voxels. Also as discussed at the end of Section 2.3.1, provided that the voxel kernel is similar to the imaging kernel $W_i$, filtering of the samples can be ignored without too many adverse effects. This allows the conditional probability distribution, $\rho_{C|S}(\mathbf{c}|\mathbf{s})$, to be expressed as

$$\rho_{C|S}(\mathbf{c}|\mathbf{s}) = \prod_{k=1}^{N} \rho_{\acute{C}_k|\check{C}_k}(\acute{c}_k(\mathbf{s}_k)|r_{\zeta_k(\mathbf{s}_k)}(\theta_k)), \tag{4.8}$$

where $\mathbf{s}_k$ are the states of voxels $\mathbf{S}_k$ located along the $k^{\text{th}}$ extended pixel ray, $\zeta_k(\mathbf{s}_k)$ is the index of the nearest opaque voxel along the $k^{\text{th}}$ extended pixel ray, and $r_{\zeta_k(\mathbf{s}_k)}(\theta_k)$ is the radiance of that voxel in the direction of the $k^{\text{th}}$ sensor element. Since the perturbed pixel positions depend on which voxel in $\mathbf{S}_k$ is nearest, the perturbed pixel intensities $\acute{c}_k(\mathbf{s}_k)$ are also a function of $\mathbf{s}_k$.

The term $\rho_{\acute{C}_k|\check{C}_k}(\acute{c}_k(\mathbf{s}_k)|r_{\zeta_k(\mathbf{s}_k)}(\theta_k))$ in Eq. 4.8 represents the probability distribution of obtaining the measured pixel intensity $\acute{c}_k(\mathbf{s}_k)$, given the ideal pixel intensity equals

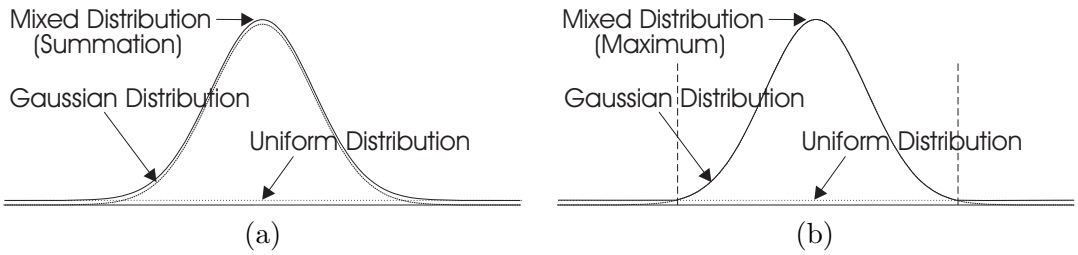


**Figure 4.1** (a) The probability distribution of the difference between the observed image intensity, $\acute{c}_k(\mathbf{s}_k)$, and the ideal image intensity, $\check{\mathbf{c}}'_k$, can be expressed using a mixed probability model, that is the weighted summation of two independent distributions. The first gaussian distribution explains the majority of the variations due to sensor noise and small modelling errors. The second uniform distribution accounts for outlying points, caused by occasional large modelling errors. (b) If the probability of outliers is reasonably small, the mixed model can be closely approximated as a maximisation over the two component distributions rather than a summation. Theoretically, this must be scaled appropriately to ensure the integral over the resulting distribution equals one. However, in practice the scale factor will be close to unity, and will not effect the MAP estimate, so can be ignored.

$r_{\zeta_k(\mathbf{s}_k)}(\theta_k)$. This probability distribution is a function of the image noise and modelling errors. This distribution can be modelled as a linear combination of two underlying distributions caused by different processes. Jaynes [2003], chapter 21, calls this a "two-model model" which is a mixture of a model that accounts for the regular observations and a second model which explains outliers.

Using the mixed model approach, the probability distribution is given by

$$\rho_{\acute{C}_k|\check{C}'_k}(\acute{c}_k(\mathbf{s}_k)|r_{\zeta_k(\mathbf{s}_k)}(\theta_k)) = (1-\nu)\rho_{\text{image}}(\acute{c}_k(\mathbf{s}_k)|r_{\zeta_k(\mathbf{s}_k)}(\theta_k)) + \nu\rho_{\text{model}}(\acute{c}_k(\mathbf{s}_k)|r_{\zeta_k(\mathbf{s}_k)}(\theta_k)), \tag{4.9}$$

where $\rho_{\text{image}}$ is the probability density function (PDF) of the image noise plus any small modelling errors, $\rho_{\text{model}}$ is the PDF of any outliers caused by occasional modelling errors, and $\nu$ is the probability of outlier observations occurring. As discussed in Section 2.2.4, the distribution of image noise can usually be closely approximated by a robust Gaussian function. Modelling errors, on the other hand, may cause significant variations in the observed pixel intensities from the ideal predicted intensity. This can be approximated using a Gaussian with a large variance, or a uniform distribution across the range of recordable pixel intensities.

As shown in Fig. 4.1, the resulting mixed distribution can be closely approximated as a weighted maximum of the two component distributions, rather than a summation. Assuming Gaussian image noise with variance $\sigma^2$, and using $\lambda_p = \nu\rho_{\text{model}}$ to represent the uniform PDF of the modelling errors, the individual probability terms are given by

$$\rho_{\acute{C}_k|\check{C}'_k}(\acute{c}_k(\mathbf{s}_k)|r_{\zeta_k(\mathbf{s}_k)}(\theta_k)) = \max\left(\frac{1}{\sigma\sqrt{2\pi}}\exp\frac{-(\acute{c}_k(\mathbf{s}_k) - r_{\zeta_k(\mathbf{s}_k)}(\theta_k))^2}{2\sigma^2}, \lambda_p\right). \tag{4.10}$$

Substituting Eq. 4.10 into Eq. 4.8, and using the resulting expression in Eq. 4.1,

the MAP scene estimate can finally be expressed as

$$S_{\text{MAP}}(\mathbf{c}) = \arg\max_{\mathbf{s}} \left[ \prod_{k=1}^{N} \max \left( \exp \frac{-(\acute{c}_k(\mathbf{s}_k) - r_{\zeta_k(\mathbf{s}_k)}(\theta_k))^2}{2\sigma^2}, \lambda_p \sigma \sqrt{2\pi} \right) \times \rho_S(\mathbf{s}) \right], \tag{4.11}$$

where the constant $1/(\sigma\sqrt{2\pi})$ has been removed from the expression, since it does not affect the MAP estimate.

By taking logarithms of each side and negating, this can alternatively be described as a summation, giving

$$\begin{aligned} S_{\text{MAP}}(\mathbf{c}) &= -\arg\max_{\mathbf{s}} \left[ \sum_{k=1}^{N} \max \left( \frac{-(\acute{c}_k(\mathbf{s}_k) - r_{\zeta_k(\mathbf{s}_k)}(\theta_k))^2}{2\sigma^2}, \log(\lambda_p \sigma \sqrt{2\pi}) \right) + \log(\rho_S(\mathbf{s})) \right], \\ &= \arg\min_{\mathbf{s}} \left[ \frac{1}{2\sigma^2} \sum_{k=1}^{N} \min \left( (\acute{c}_k(\mathbf{s}_k) - r_{\zeta_k(\mathbf{s}_k)}(\theta_k))^2, \lambda_e \right) - \log(\rho_S(\mathbf{s})) \right], \end{aligned} \tag{4.12}$$

where

$$\lambda_e = -2\sigma^2 \log(\lambda_p \sigma \sqrt{2\pi}), \tag{4.13}$$

is the robustness parameter.

Instead of expressing the data error in Eq. 4.12 as a summation over the set of image pixels, it can instead be expressed as a summation over the scene voxels, giving

$$S_{\text{MAP}}(\mathbf{c}) = \arg\min_{\mathbf{s}} \left[ \frac{1}{2\sigma^2} \sum_{j=1}^{M} \sum_{k \in \{k:\zeta_k(\mathbf{s}_k)=j\}} \min \left( (\acute{c}_k(\mathbf{s}_k) - r_j(\theta_k))^2, \lambda_e \right) - \log(\rho_S(\mathbf{s})) \right], \tag{4.14}$$

where $j$ are the voxel indices and $\{k : \zeta_k(\mathbf{s}_k) = j\}$ is the set of pixels for which $\zeta_k(\mathbf{s}_k) = j$. This can more conveniently be expressed in terms of voxel visibilities $\Omega_j(\mathbf{s})$, where $\Omega_j(\mathbf{s})$ is the set of pixels which can observe voxel $j$, giving

$$S_{\text{MAP}}(\mathbf{c}) = \arg\min_{\mathbf{s}} \left[ \sum_{j=1}^{M} E_j(s_j, c, \Omega_j(\mathbf{s})) - \log(\rho_S(\mathbf{s})) \right], \tag{4.15}$$

where

$$E_j(s_j, c, \Omega_j(\mathbf{s})) = \begin{cases} \dfrac{1}{2\sigma^2} \displaystyle\sum_{k \in \Omega_j(\mathbf{s})} \min \left( (\acute{c}_k(\mathbf{s}_k) - r_j(\theta_k))^2, \lambda_e \right) & \text{if } \alpha_j = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{4.16}$$

and $\alpha_j$ is the opacity of the $j^{\text{th}}$ voxel.

This states that the most likely or probable estimate of $S$ is the one which minimises the right hand objective function. This function consists of two terms: the first is an error function between the estimated and actual image intensities weighted by one over

twice the noise variance, while the second is the unconditional negative log probability of the estimate obtained from prior information. A greedy approach for minimising this objective function is presented in Section 4.6. The effect of the second term can be reduced by either increasing the number of data points or reducing the image noise. By ignoring the second term entirely, the maximum likelihood (ML) estimate of $S$ is obtained. A novel approach for minimising this slightly simpler ML objective function is investigated in Sections 4.2 to 4.5.

## 4.2 PIXEL RAY ASSIGNMENT

Assuming a complete voxel based scene estimate, the scene reconstruction problem can be expressed as an assignment problem, where the objective is to assign at least one opaque voxel along every extended pixel ray, so that the objective function given in Eq. 4.15 is minimised. This problem is non-trivial, as the objective function is a complex function of the voxel opacities and radiances.

To simplify the optimisation problem, information or assumptions about the scene visibilities can be used to reduce the interaction between voxel parameters. This enables the data error terms, $E_j(s_j, c, \Omega_j(\mathbf{s}))$, to be expressed as $E_j(s_j, c, \Omega_j)$, allowing them to be calculated independently for each voxel. The difficulty with this approach is that scene visibilities are usually unknown prior to reconstruction and so must also be estimated, leading to additional complications. Care should also be taken to ensure the assignment of opaque voxels is consistent with the visibility assumptions, otherwise the calculated error terms will be incorrect.

To minimise the objective function given in Eq. 4.15, the estimated radiance, $r_j(\theta)$, and opacity, $\alpha_j$, of each voxel must be chosen so as to minimise the sum of the overall data error term, $\sum_{j=1}^{M} E_j(s_j, c, \Omega_j(\mathbf{s}))$, and the negative log prior probability term, $-\log(\rho_S(\mathbf{s}))$. For a given estimate of the voxel opacity, the values of $r_j(\theta)$ which minimise the objective function can usually be closely approximated by independently minimising the voxel error terms, $E_j(s_j, c, \Omega_j(\mathbf{s}))$. Assuming Lambertian reflection, where $r_j(\theta) = r_j$, and ignoring the effect of outliers, this is achieved by setting $r_j$ equal to the mean of the observed pixel intensities in $\Omega_j(\mathbf{s})$ if $\alpha_j = 1$, and equal to zero otherwise. If the scene radiances are required to be estimated more accurately, the effect of outliers can be accounted by minimising the full error term given in Eq. 4.16. An efficient technique for doing this is presented by Jonsson and Felsberg [2005].

Using $\bar{\mu}_j(c, \Omega_j(\mathbf{s}))$ to represent the mean intensity of the perturbed pixels observing $s_j$, and $|\Omega_j(\mathbf{s})|$ to represent the number of pixels observing $s_j$, the voxel radiances can be expressed as

$$r_j = \begin{cases} \bar{\mu}_j(c, \Omega_j(\mathbf{s})) & \text{if } \alpha_j = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{4.17}$$

where

$$\bar{\mu}_j(c, \Omega_j(\mathbf{s})) = \frac{1}{|\Omega_j(\mathbf{s})|} \sum_{k \in \Omega_j(\mathbf{s})} \acute{c}_k(\mathbf{s}_k). \tag{4.18}$$

Substituting Eq. 4.17 into Eq. 4.16, the error term, $E_j(s_j, c, \Omega_j(\mathbf{s}))$, is given by

$$E_j(s_j, c, \Omega_j(\mathbf{s})) = \begin{cases} \dfrac{1}{2\sigma^2} \displaystyle\sum_{k \in \Omega_j(\mathbf{s})} \min\left((\acute{c}_k(\mathbf{s}_k) - \bar{\mu}_j(c, \Omega_j(\mathbf{s}))^2, \lambda_e\right) & \text{if } \alpha_j = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{4.19}$$

This defines the projected error of each voxel, as a function of its opacity $\alpha_j$, visibility $\Omega_j(\mathbf{s})$, and the camera data. The problem with minimising the overall objective function, is that the visibility of each voxel depends on the state of numerous other voxels within the scene. This results in a complex, non linear error term, which is extremely difficult to optimise.

To simplify the optimisation, the problem of finding the MAP scene estimate is initially investigated under the simplified assumption that all surface points are fully visible, independent of the state of other points within the scene. This allows the probability of obtaining the observed pixel data corresponding with any voxel to be easily calculated independently of the other voxels. With this assumption the visibility term $\Omega_j(\mathbf{s})$ can be expressed as

$$\Omega_j(\mathbf{s}) = \grave{\Omega}_j, \tag{4.20}$$

where $\grave{\Omega}_j$ is the full set of pixels whose extended rays pass through voxel $j$. This allows the error term $E_j(s_j, c, \Omega_j(\mathbf{s}))$ to be calculated independently for each voxel as a function of its opacity and the camera data.

To further simplify the analysis, it is also initially assumed that the prior likelihood of all scenes are equal. This is equivalent to finding the Maximum Likelihood (ML) estimate of the scene. With these approximations the ML scene estimate is given by

$$\begin{aligned} S_{\mathrm{ML}}(\mathbf{c}) &= \arg\min_{\mathbf{s}} \sum_{j=1}^{M} E_j(s_j, c, \Omega_j), \\ &= \arg\min_{\mathbf{s}} \sum_{j \in P} E_j^o(c, \Omega_j), \end{aligned} \tag{4.21}$$

where $P$ is a complete set of opaque voxels and $E_j^o(c, \Omega_j)$ is the data error of voxel $j$ being opaque given the current visibility estimate.

This can be expressed as a pixel ray assignment problem, where the objective is to assign at least one opaque voxel along every extended pixel ray, so that the sum of assigned error terms is minimised. By describing each extended pixel ray as a set of voxels, the pixel ray assignment problem can be stated more formally as follows: 'Given a set of values that is divided into a number of subsets, assign at least one element in every subset, such that the overall sum of the assigned values is a minimum'. The

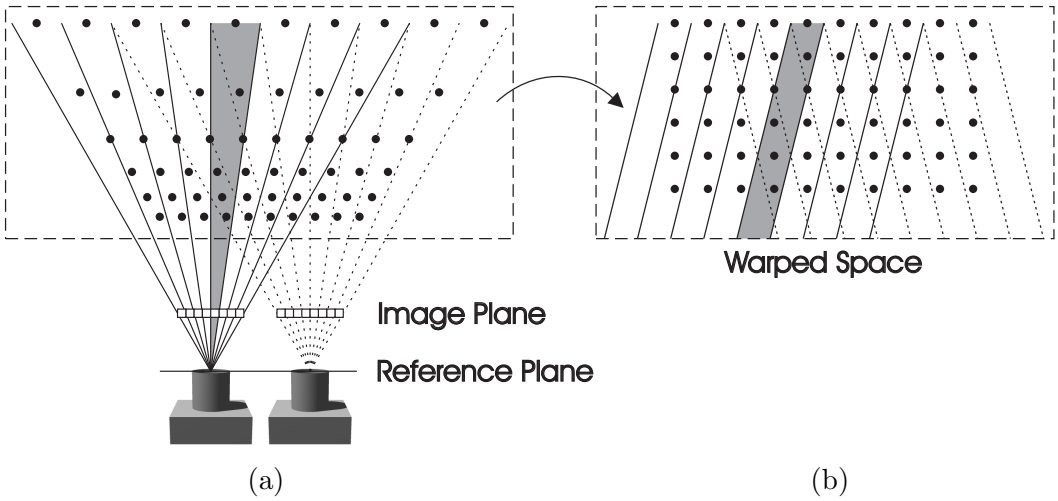


**Figure 4.2** (a) With a planar camera configuration, the sample or voxel spacing can be chosen so that all voxel kernels project to a constant sized area, or footprint, in each image. To simplify the mapping, voxels are positioned on planes that are uniformly spaced in inverse depth. This corresponds well with human perception, as it places greater importance on nearby objects, which are visually more significant. (b) With samples positioned in this way, the pixel rays from a given camera will all be parallel if the scene space is warped so that the voxels lie on a regular grid.

solution to this problem will give the most likely estimate of the scene under the previous assumptions.

To implement this, a planar camera configuration is used, where all the imaging cameras lie on a reference plane and face perpendicular to it. With the cameras positioned in this way, the width and height of all pixel rays passing through any scene point will be equal. This ensures the imaging convolution kernel $W_i(x, y, Z, u, v, w)$, described in Section 2.3.4, remains relatively constant between images, so long as the width of the kernel in the $Z$ direction is not significantly larger than its width in the $X$ and $Y$ directions. By setting the voxel spacing in the $X$ and $Y$ direction equal to the separation between pixel rays, the voxel kernel will be a close approximation of the imaging kernel, avoiding the need for filtering the samples.

To further simplify the mapping, the voxels are spaced inversely proportional to depth in the $Z$ direction. This has the useful property that, if the scene space is warped so that the voxels lie in a regular grid, the resulting pixel rays from a given camera will all be parallel; see Fig. 4.2(a) and (b). This is important computationally, as the algorithm can more easily be optimised for efficiency on a regular grid.

Given a set of voxels, the data error, $E_j^o(c, \Omega_j)$, for each voxel being opaque is obtained by calculating the sum of square errors of the perturbed pixel intensities, $\acute{c}_k(\mathbf{s}_k)$, that project to that point. These intensities are found by interpolating the obtained pixel data, $c_k$. For computational efficiency, this is performed using linear interpolation.

The set of voxels is then divided up into a number of subsets corresponding to the extended pixel rays from each pixel. Each subset is defined as the set of voxels that

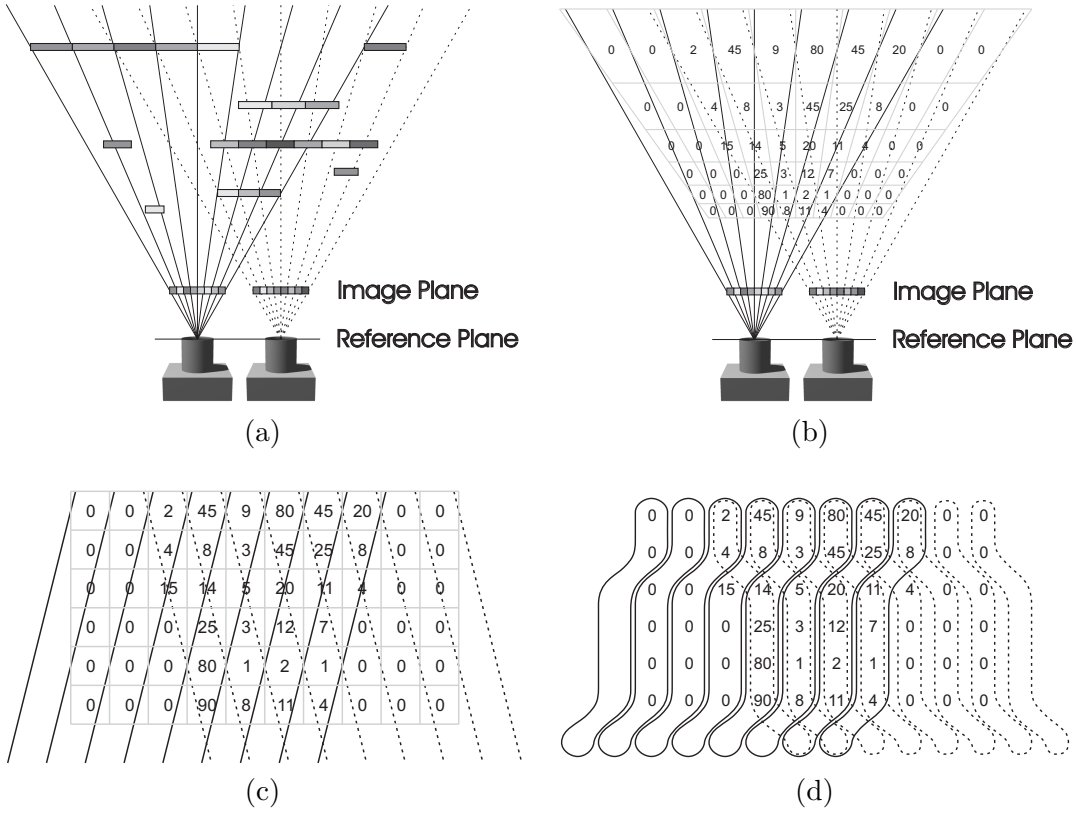


**Figure 4.3** (a) To form an estimate of the scene, intensity information about the scene is acquired by a number of images. Given this data, the reconstruction problem can be formulated as determining the most likely estimate of the scene. By making some assumptions about the scene visibilities, this can be expressed as a pixel ray assignment problem, where the objective is to assign at least one opaque voxel along every extended pixel ray so that the sum of the values associated with each voxel is minimised. (b) Using a voxel based scene model, the projected Mean Square Error (MSE) associated with each voxel being opaque is computed by interpolating between the given pixel samples. (c) To simplify the mapping and improve the efficiency of the algorithm, the scene space can be warped so that the voxels lie in a regular grid. With the voxels evenly spaced in inverse depth, the resulting pixel rays from each camera will be parallel. (d) To express the reconstruction problem as an assignment problem, the set of scene voxels is divided up into a number of subsets corresponding with the extended pixel rays from each pixel.

are intercepted by the extended pixel ray. This is demonstrated in Fig. 4.3. The most likely scene estimate is then found by assigning at least one opaque voxel within each subset, so that the sum of projected errors is as small as possible.

Determining which voxels to assign as opaque to minimise Eq. 4.21 is a difficult problem. Because the subsets are not disjoint, the overall minimum cannot be found by minimising each subset independently. A brute force approach, where every possible combination is tried, is also infeasible, since the number of combinations will be enormous. To help solve this problem, an efficient heuristic approach for finding an approximate solution is presented.

## 4.3 ASSIGNMENT ALGORITHM

To find a near optimal solution to the pixel ray assignment problem, a new greedy type algorithm is presented. This progressively assigns voxels as opaque until every subset contains at least one opaque voxel. To achieve this, a benefit measure is first calculated for each voxel. This is a function of the voxel's projected data error, the current set of assignments, as well as the data error of all other voxels that are within the same subsets. To simplify the description of the algorithm, the term 'cost' will be used to describe the projected data error, $E_j^o(c, \Omega_j)$ of each voxel. The algorithm then assigns voxels as opaque using a greedy selection process, where at each iteration the voxel with maximum benefit is assigned as opaque. Having made an assignment, the benefit of other affected voxels is updated and the process repeated until at least one opaque voxel has been assigned within every subset.

The benefit of a voxel is defined to be the minimum extra cost that would be incurred if that voxel is not assigned as opaque. Since every subset must contain at least one opaque voxel, then by not assigning a voxel, at least one other voxel within each subset containing that voxel must be assigned as opaque. To calculate a voxel's benefit, the cost of the voxel is subtracted from the sum of minimum costs within each subset of which the voxel is a member. These minimum subset costs must exclude the cost of the voxel whose benefit is being calculated. This has close similarities to the selection process used for branching in the algorithm proposed by Little et al. [1963] for the Travelling Salesman problem.

For subsets that already contain an assigned voxel, no additional voxel within the subset needs to be selected. Therefore, the minimum additional cost associated with that subset will be zero. Using $\psi_i$ to denote the subset of voxels corresponding to pixel ray $i$, the voxel benefits are given by

$$\text{benefit}_j = \sum_{i:\psi_i \supset j} \Big( \mathrm{U}_i(\mathbf{s}) \min_{k \in \psi_i \setminus j} \text{cost}_k \Big) - \text{cost}_j, \tag{4.22}$$

where $\setminus$ is the set minus operator, and $\mathrm{U}_i(\mathbf{s})$ is a binary function that is equal to zero if $\psi_i$ contains an assigned opaque voxel and equal to one otherwise. The term $\mathbf{s}$ is used in this expression, to refer to the current scene estimate or set of assignments.

If all subsets of a voxel already contain another opaque voxel, then the overall additional cost will be zero. This will result in a negative benefit, so long as all costs are positive definite. To ensure this is the case, a very small constant is added to each of the costs. Since the maximum benefit within any unassigned subset will always be greater than or equal to zero, a global maximum benefit less than zero implies that the scene is complete. This can be used as a stopping criterion for the algorithm, where the assignment process is repeated until the maximum remaining benefit is less than zero. The steps of the pixel ray assignment algorithm are illustrated in more detail using a simple example in Fig. 4.4.

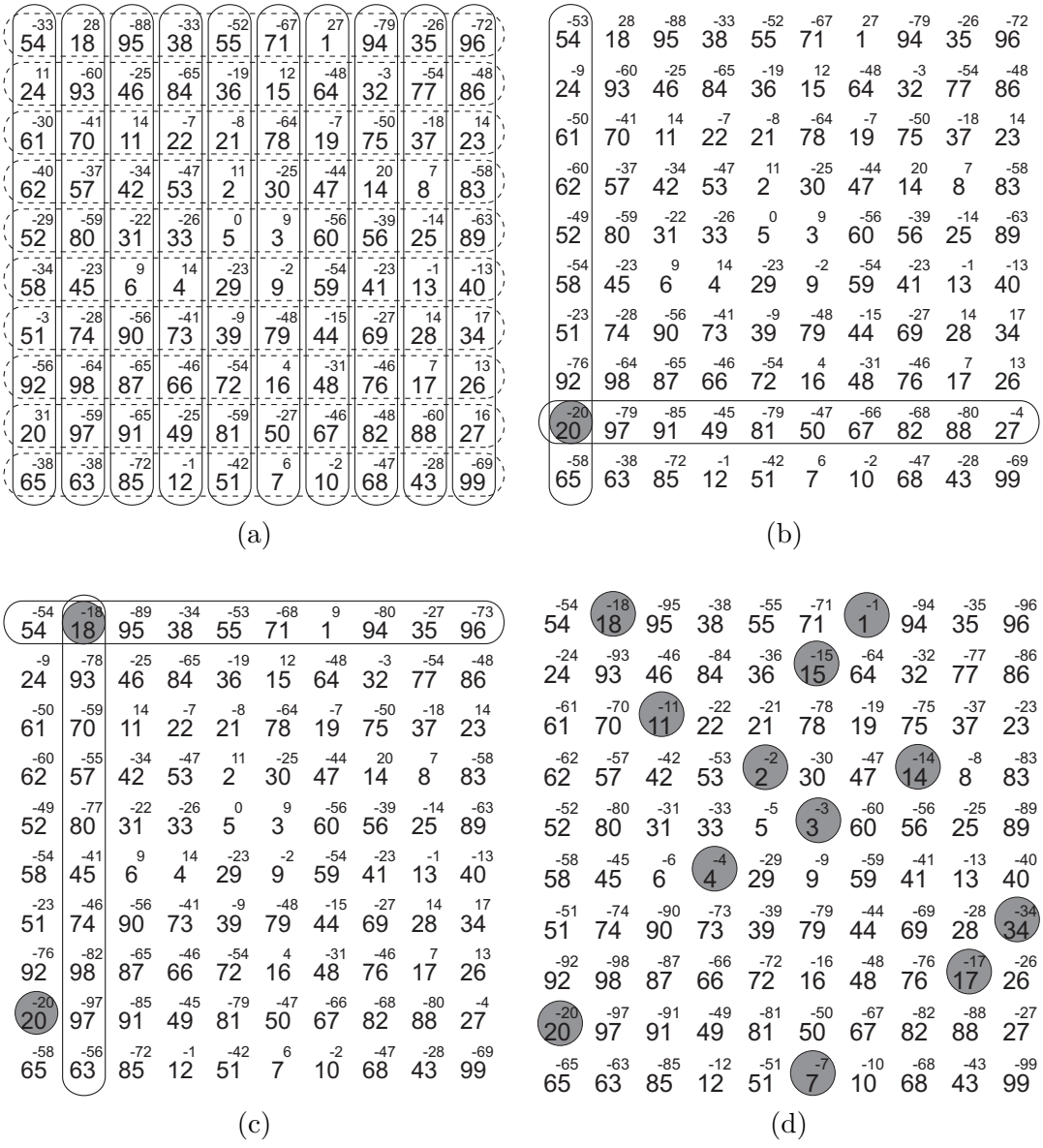


**Figure 4.4** To demonstrate the pixel ray assignment algorithm, a simple example is shown where the scene voxels are divided up into a number of subsets corresponding to the rows and columns of a matrix. The objective of the algorithm is to assign at least one voxel as opaque within every subset, so that the sum of assigned costs is a minimum. (a) To begin, a benefit measure is calculated for each voxel. This is achieved by summing together the minimum costs, excluding that of the voxel, within each subset the voxel is a member of. The cost of the voxel is then subtracted from this summation to give the benefit of the voxel. (b) Using a greedy approach, the voxel with maximum benefit is assigned as opaque. The benefit of all voxels within any subset that includes the assigned voxel are then updated. Since these subsets no longer require a voxel to be assigned as opaque, the minimum cost or benefit associated with each subset will be zero. (c) Having updated the benefits, the new voxel with maximum benefit is assigned as opaque and the process repeated. (d) This assignment process is continued until all the voxel benefits are negative. At this point every subset will contain at least one opaque voxel and the scene estimate will be complete.

### 4.3.1 Results

To demonstrate the performance of the pixel ray assignment algorithm, a synthetic test scene and set of images were generated using the ray tracing program POV-Ray [POV-Ray 2007]. The generated test scene consisted of a variety of different shaped objects and surface textures, testing the algorithm under a range of conditions. The horizontal base plane was coloured with a high frequency pattern to test the reconstruction algorithm on sharply changing intensities and repetitive patterns, as commonly occur in urban or indoor environments. While the vertical background plane consisted of low frequency intensity variations with a small amount of speckle, testing the algorithm on reasonably bland regions with subtle high frequency detail, such as occur in many outdoor scenes including grassy areas and distant trees or bushes. The collection of foreground objects were chosen to test the algorithm on a variety of different shapes and sizes, including both curved and angular objects, as well as coarse and fine detail. In this test, as with the other tests conducted in this thesis, only greyscale image data is used. Surface radiances were set in the range 0 to 255.

The use of a synthetic test set enabled the resulting scene estimate to be compared with an ideal set of scene parameters, helping to evaluate its performance. Another advantage of using a synthetic scene is that the radiometric properties of the scene and images can be set as required. To help evaluate the algorithm, the test scene was modelled using opaque surfaces and Lambertian reflectance. This corresponds with the assumptions that are made in deriving the pixel ray assignment algorithm.

To test the algorithm, five images of the scene were generated from equal spaced positions along the $X$ axis, with the cameras facing in the $Z$ direction. To simulate image noise, independent Gaussian noise was added to each image, giving a Signal to Noise Ratio (SNR) of 30 dB. The left, centre, and right most images of this sequence are shown in Fig. 4.5, along with the corresponding depth-maps as viewed from these three camera positions. These images and scene are referred to as the "shapes" test set.

When calculating the cost of each voxel using Eq. 4.19, the term $\lambda_e$ was set to infinity, corresponding with no outlying data. Finite values of $\lambda_e$ were also tried, but the projected square error of the estimated intensities and depth map steadily increased as $\lambda_e$ was reduced. The incorporation of $\lambda_e$ in the error function is really only important when prior information about the scene is included into the estimation process.

The results from the pixel ray assignment algorithm are shown in Fig. 4.6. The projected MSE of the reconstructed scene is approximately equal to 5 over most regions of the scene, corresponding closely with the variance of the image noise. However, there are significant errors in semi-occluded regions, as well as in regions where there is a large intensity variation between neighbouring pixels. In addition to projected intensity errors, the resulting depth-maps are rather poor with numerous large errors, especially in semi-occluded and textureless regions.

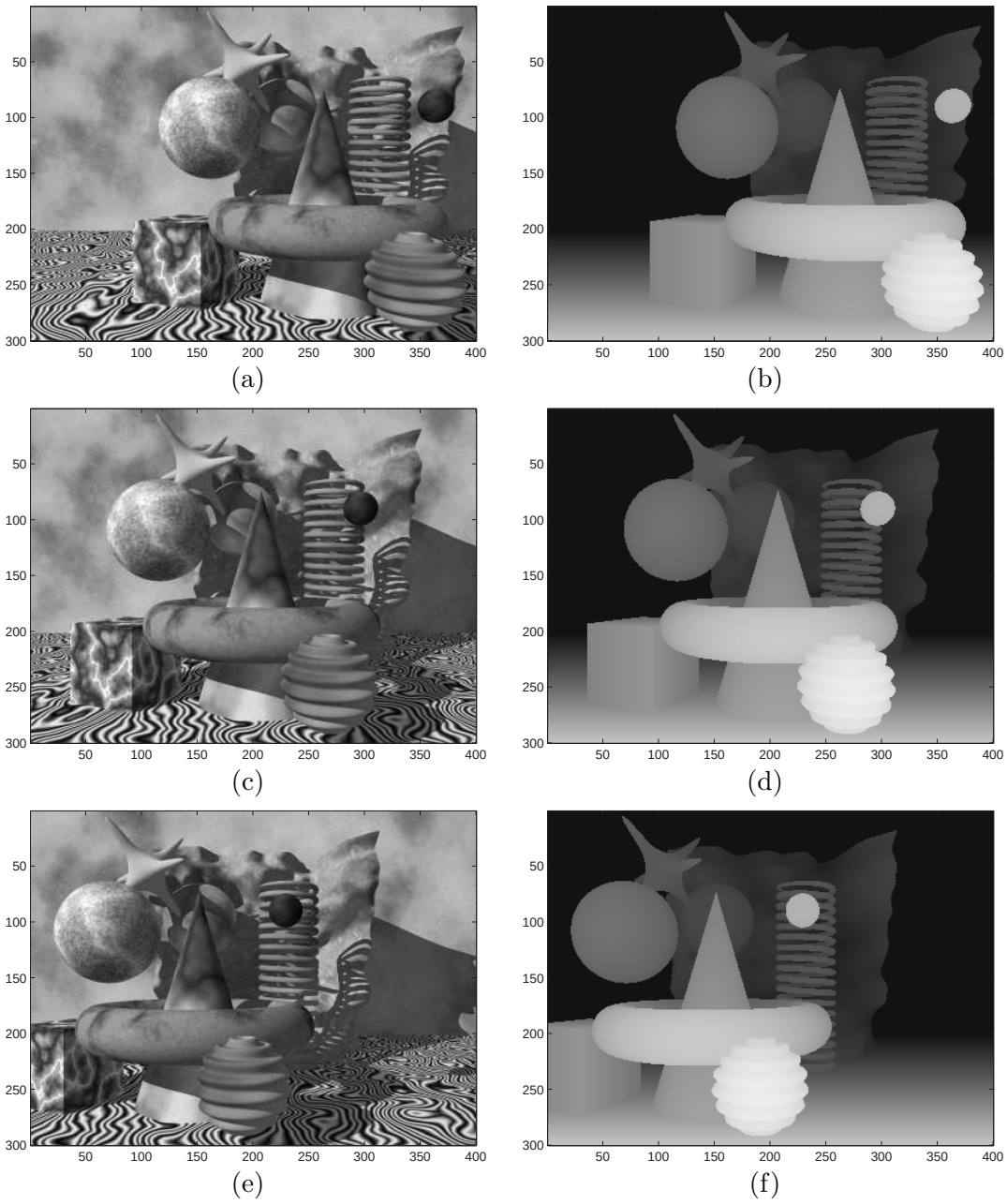


**Figure 4.5** Synthetic test scene used for testing algorithms. (a), (c), and (e) Left, centre, and right most images from sequence of five images. (b), (d), and (f) Depth-maps of the scene corresponding with these camera positions.

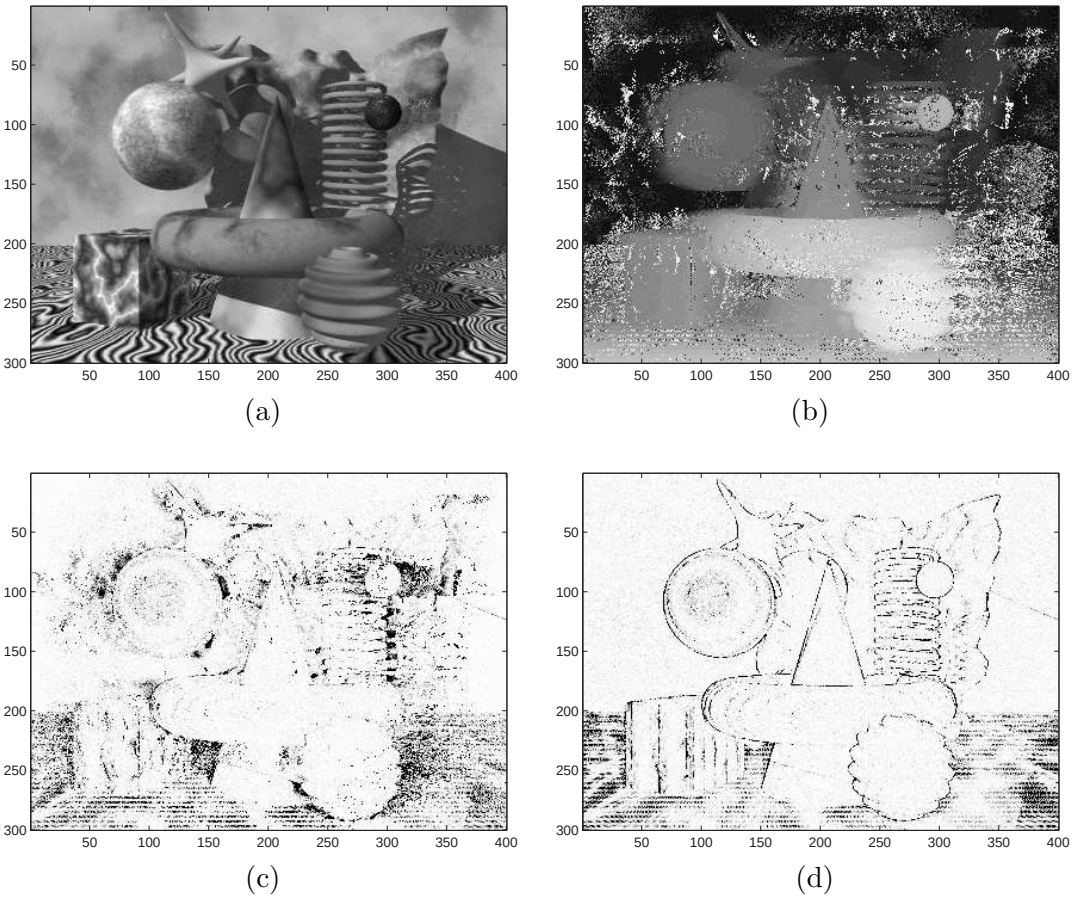


**Figure 4.6** Results of pixel ray assignment algorithm. (a) Image of reconstructed scene from central camera position. As shown, the majority of observable errors appear in semi-occluded regions. (b) Depth-map of reconstructed scene from central camera position. This contains a large number of errors, especially in textureless or semi-occluded regions. (c) Square error between reconstructed central image and original image. This show more clearly the distribution of errors in semi-occluded regions, as well as highlighting the occurrence of errors in regions with large intensity variations between neighbouring pixels. (d) Projected square error of ideal voxel parameters using binary opacities. Comparing with Fig. 4.5(c) it is observed that most errors occur around depth discontinuities and in regions with a high intensity variation between neighbouring pixels. These variations are caused by differences in the convolution kernel between images, illustrated in Figures 2.9 and 6.7, as well as approximations in the binary opacity voxel model, shown in Fig. 2.13.

With this test set, the resulting scene estimate gives a sum of assigned costs equal to $53.89 \times 10^6$. Although this is large compared with the sum of the image noise, which equals $3.37 \times 10^6$, the sum of assigned costs for an ideal set of voxel parameters equals $190.46 \times 10^6$. Therefore, the main problem with the pixel ray assignment algorithm is not minimising the cost function but that the cost function itself is a poor measure of the likelihood of the scene.

One of the main reasons for this is the false assumption that all opaque voxels are fully visible. Although this assumption is true for the majority of surface voxels, there are a large number that are only visible in a subset of the images.

The other major causes of errors are variations in the convolution kernel between images, discussed in Section 2.3.4, as well as approximations in the binary opacity voxel model, shown in Fig. 2.13. Even with an ideal set of scene parameters, there will be significant variations in the modelled projected intensities from what is actually observed. These errors occur mainly at depth discontinuities and regions of high intensity variation as shown in Fig. 4.6(d). For this test set, the projected square error of the ideal binary opacity voxel parameters was $38.92 \times 10^6$. Fortunately, in most instances these errors are not perceptible to the human eye, as they are the result of subtle sub-pixel shifting of the observed intensities. A novel pixel dissimilarity measure for dealing with these variations is presented in Section 6.4.

## 4.4 VISIBILITY UPDATING

To improve the results of the pixel ray assignment algorithm, the visibility estimate must be improved in occluded or semi-occluded regions, so that the $\text{cost}_j(c, \Omega_j)$ can be calculated more accurately. Unfortunately, voxel visibilities are unknown prior to reconstruction, and the resulting model visibilities will depend on the final scene estimate.

A common approach is to try and estimate a voxel's visibility by comparing or filtering the observed pixel intensities from within the set of pixels $\grave{\Omega}_j$. A simple and effective way of doing this is to use a visibility mask, or take the best 50% of matches [Satoh and Ohta 1996]. With a linear arrangement of cameras, this is commonly achieved by taking the best half sequence [Kang et al. 2001]. Such an approach works reasonably well for reconstructing a depth-map relative to one of the central cameras. Problems obviously arise in regions that are occluded in more than half the images. Also, by only using half the images, the probability of a false match is increased. This can lead to voxels being estimated as opaque, even if they do not correspond with the data in all images.

To generalise the pixel intensity filtering approach for estimating a voxels visibility, the data error within various subsets of the pixels in $\grave{\Omega}_j$ is calculated and the minimum of these errors then taken as the overall data error. For a linear array of cameras, as presented here, these subsets can be chosen to correspond with overlapping sequences

of the images from left to right. Calculation of the data error can be implemented efficiently using a 1D version of the box filtering technique described by Sun [1997] for calculating the variance within overlapping subregions of an image.

Using overlapping subsets of three images from left to right, the pixel ray assignment algorithm was retested on the previous data set to determine the effect this would have on the resulting estimate. As shown in Fig. 4.7(a) and (b), the projected intensity and depth-map from the central camera position is improved in semi-occluded regions. However, overall the depth-map is significantly noisier, due to an increase in the number of false matches. When viewed from the left most camera position, as shown in Fig. 4.7(c), the estimated scene no longer corresponds closely with the image data. The depth-map from this camera position is also noticeably poorer than from the central camera position. The reason for this is that the central camera image is the only image that is a member of all subsets. Therefore, a low voxel cost ensures a voxel is consistent with the central image but not necessarily with any other image.

To improve the estimation of voxel visibilities, and ensure the visibilities are consistent with the reconstructed scene, a novel iterative approach for estimating the voxel visibilities is proposed. Beginning with the assumption of full visibility, voxel visibilities are progressively updated based on the current set of assignments. After each assignment, the visibility and cost of affected voxels is updated, along with the benefit of any affected voxel.

Using this iterative approach for estimating voxel visibilities, the pixel ray assignment algorithm was retested on the synthetic test set. Results are shown in Fig. 4.8. Comparing Fig. 4.8(a) with Fig. 4.6(c), the projected square error is $23.18 \times 10^6$ and $37.86 \times 10^6$ respectively. Visibility updating therefore improves the scene estimate, however numerous errors still occur in semi-occluded regions. The accuracy of the resulting depth-map is also significantly worse than that obtained under the full visibility assumption.

The problem with the pixel ray assignment algorithm is that the calculated benefit of a voxel may decrease as the scene visibilities are updated. Therefore, some voxels which initially appear highly beneficial may be assigned as opaque even though they are not beneficial under the final visibility estimate. As a consequence, the resulting reconstruction is unlikely to correspond particularly well with the observed data.

## 4.5 GREEDY ALGORITHM

To improve the assignment process when updating visibilities, a novel alternative greedy algorithm is presented. This is similar to the pixel ray assignment algorithm, except that the voxel benefits are defined differently.

Instead of calculating a voxel's benefit by subtracting the cost of the voxel from the sum of minimum costs within each subset containing the voxel, voxel benefits are

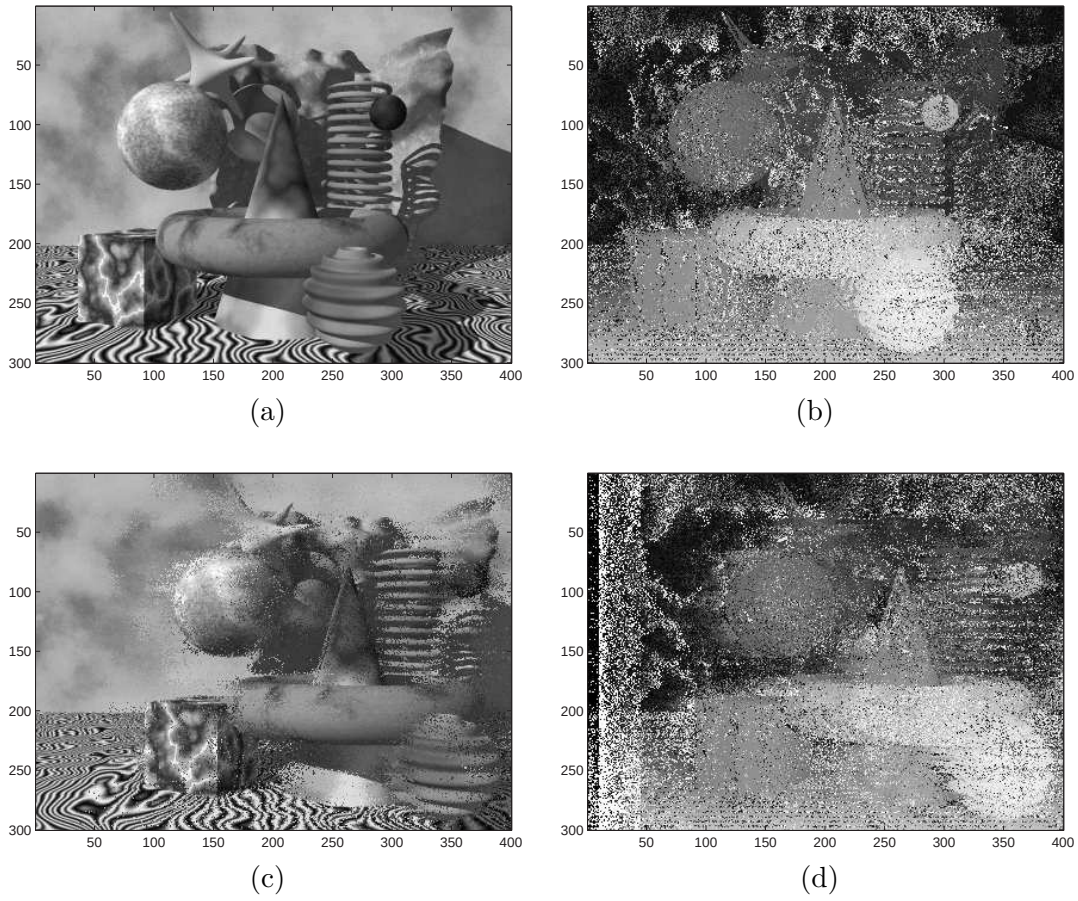


**Figure 4.7** Results of pixel ray assignment algorithm using best half sequence. (a) Image of reconstructed scene from central camera position. As shown, this corresponds closely with original data. (b) Depth-map from central camera position. Reconstruction is more accurate in semi-occluded regions than with full visibility assumption but not so good overall. (c) Image of reconstructed scene from left-most camera position. This does not correspond as well with the original data. (d) Depth-map from left-most camera position. The mean square error is higher than with full visibility assumption in both semi-occluded and fully visible regions.

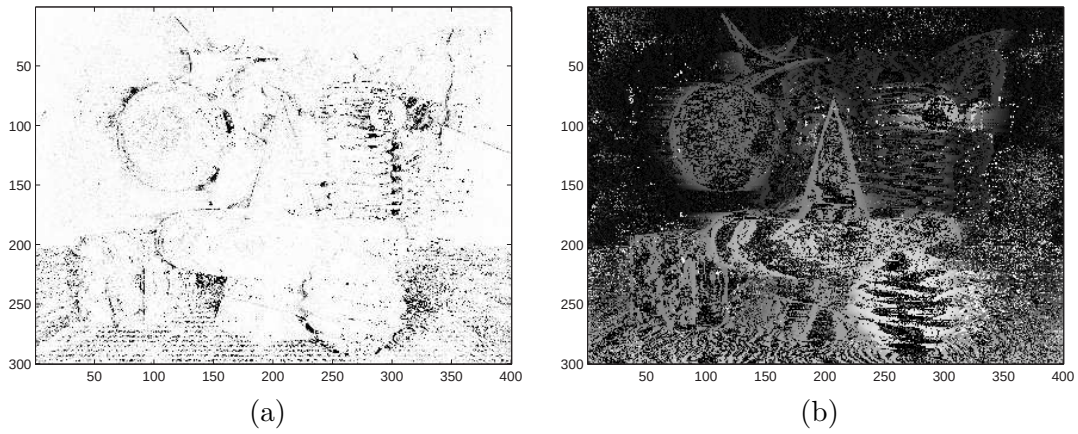


**Figure 4.8** Results of the pixel ray assignment algorithm with visibility updating. (a) Square error between reconstructed central image and original image. (b) Depth-map from central camera position.

defined as being equal to one over the cost of the voxel multiplied by the number of unassigned subsets containing the voxel, giving

$$\text{benefit}_j = \frac{\sum_{i:\psi_i \supset j} U_i(\mathbf{s})}{\text{cost}_j}. \tag{4.23}$$

The greedy benefit measure should still result in a reasonable solution to the pixel ray assignment problem, while ensuring voxel benefits remain relatively constant or improve as the visibility estimate is refined.

By assuming voxels are fully visible in each direction unless explicitly occluded, the projected square error associated with a voxel being opaque will never decrease as the reconstruction progresses. Therefore, assigned voxels will remain likely, even if they are subsequently determined to be occluded. The full visibility assumption also encourages more visible surface regions to be reconstructed first, since these will tend to match the camera data more closely under this assumption. As a consequence, the visibility of assigned voxels is unlikely to change. These two properties help prevent the assignment of false voxels, aiding the convergence of the algorithm to a strong local minimum.

To compare the performance of the greedy algorithm with the pixel ray assignment algorithm, the greedy algorithm was tested on the synthetic test set, both with and without visibility updating. Results are shown in Fig. 4.9. As can be seen in Fig. 4.9(a) and (b), the greedy algorithm produces similar results to the pixel ray assignment algorithm when visibilities are not updated. The resulting sum of assigned costs equals $57.48 \times 10^6$, which is slightly higher than with the pixel ray assignment algorithm but still significantly less than that of the ideal binary opacity voxel parameters. With visibility updating, the greedy algorithm performed substantially better than the pixel ray assignment algorithm. As shown in Fig. 4.9(c), the resulting projected square error from the central camera position was very small over most of the image, with a few errors in regions with a large intensity variation between neighbouring voxels. Similar errors were observed at the other camera positions. The overall projected square error was significantly less than that of the ideal binary opacity voxel parameters.

Although the greedy algorithm succeeds in minimising the projected square error, the resulting scene estimate is significantly different from the ideal binary opacity voxel parameters. This is apparent by comparing the depth-map from the central camera position, shown in Fig. 4.9(d), with the ideal depth-map, shown in Fig. 4.5(d). This highlights the importance of prior information, since a reliable estimate of the scene cannot be obtained by minimising the data term in Eq. 4.15 alone.

## 4.6 PRIOR INFORMATION

Although the basic greedy algorithm does a good job of minimising the projected square error, the resulting scene estimate is unlikely to correspond closely with the actual scene.

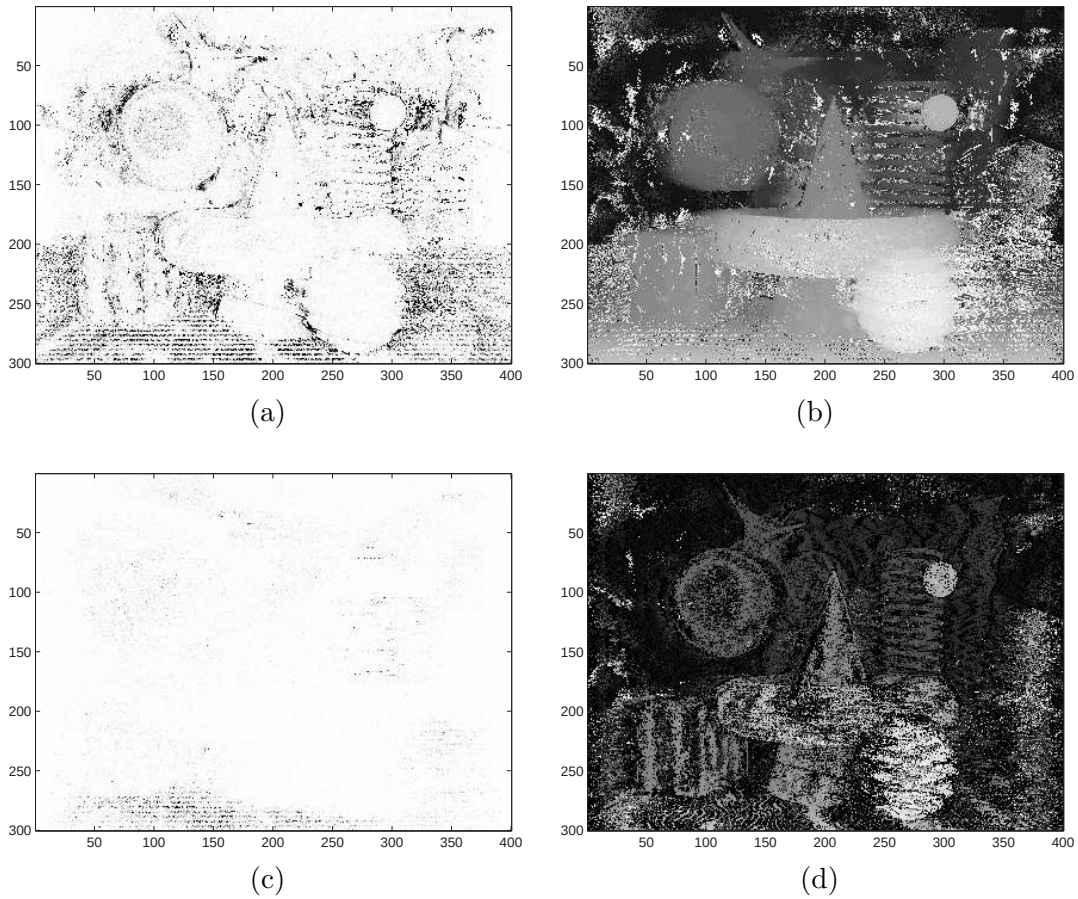


**Figure 4.9** Results of the greedy algorithm. (a) Without visibility updating, the resulting square error between the reconstructed central image and original image, is similar to that obtained by the pixel ray assignment algorithm. (b) The depth-map from central camera position is also similar. (c) With visibility updating, the square error between the reconstructed central image and original image is significantly reduced, and is noticeably less than that obtained by the pixel ray assignment algorithm. (d) Despite improvements in the projected square intensity error, the depth-map from each camera position is similar in quality to that obtained using the assignment algorithm.

The reason for this is that, because of noise and ambiguities in the inverse mapping, a large number of potential scenes are likely to correspond well with the image data. To improve the scene estimate, additional prior information about the scene must be used.

By expressing the log prior probability of the scene as a summation of terms corresponding with each voxel, prior information can be incorporated into the cost functions. From standard probability theory, the overall prior scene probability can be expressed using the chain rule for probabilities, as

$$\rho_S(\mathbf{s}) = \prod_{j=1}^{M} \rho_{S_j|S_{k<j}}(s_j|s_{(j-1)}, \ldots, s_1), \tag{4.24}$$

where $M$ is the number of elements of $S$. Substituting this expression into Eq. 4.15, the MAP estimate can alternatively be expressed as

$$\begin{aligned} S_{\text{MAP}}(\mathbf{c}) &= \arg\min_{\mathbf{s}} \left[ \frac{1}{2\sigma^2} \sum_{j=1}^{M} E_j(s_j, c, \Omega_j(\mathbf{s})) - \sum_{j=1}^{M} \log(\rho_{S_j|S_{k<j}}(s_j|s_{(j-1)}, \ldots, s_1)) \right], \\ &= \arg\min_{\mathbf{s}} \sum_{j=1}^{M} \left[ \frac{1}{2\sigma^2} E_j(s_j, c, \Omega_j(\mathbf{s})) - \log(\rho_{S_j|S_{k<j}}(s_j|s_{(j-1)}, \ldots, s_1)) \right]. \end{aligned} \tag{4.25}$$

This suggests that prior information can be incorporated by adding a negative conditional log probability term to the cost of each voxel. The problem with this approach is that the log probability terms depend not only on the state of numerous other voxels but also on the chosen ordering of the voxels. Consequently, minimising this function is rather difficult.

Using the greedy approach, the ordering of the prior log probability terms is chosen to correspond with the order of voxel assignments. This enables the prior terms, calculated using Eq. 4.25, to be equal to the conditional probability that each voxel is opaque given the current set of assignments.

One of the most useful and commonly used priors is the fact that most scenes consist of several piece-wise continuous radiating surfaces, rather than a cloud of point sources. This prior can be applied to the greedy algorithm by favouring the assignment of opaque voxels that are neighbours of voxels which have already been assigned as opaque. This is achieved by adding a neighbourhood similarity term, $N_j(\mathbf{s})$, to the cost of each voxel, which is a function of the number and location of assigned voxels within the neighbourhood of voxel $j$.

Another related constraint that applies with a planar camera configuration, is that if a surface is visible in one of the cameras it is likely to be visible to the other cameras. Therefore, preference should be given to scenes where the average visibility of assigned voxels is greater. This is accomplished by minimising the number of voxels assigned as opaque. To achieve this, a visibility term, $V_j(\mathbf{s})$, is added to the cost of each voxel.

This is a function of the number of unassigned pixel rays passing through each voxel and favours the assignment of voxels that are members of a larger number of unassigned subsets.

Treating $N_j(\mathbf{s})$ and $V_j(\mathbf{s})$ as pseudo probabilities, and using the expression for the MAP estimate given in Eq. 4.15, the modified voxel costs are defined as

$$\text{cost}_j(c, \mathbf{s}) = E_j^o(c, \Omega_j) - \log(V_j(\mathbf{s})N_j(\mathbf{s})). \tag{4.26}$$

The neighbourhood terms, $N_j(\mathbf{s})$, were calculated by convolving the scene opacities with a neighbourhood window function, $W_N$, and using the result as the index to a lookup table, $P_N$. For these experiments, the neighbourhood window function was defined as

$$W_N = \underset{Z=-1}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}} \quad \underset{Z=0}{\begin{bmatrix} 4 & 4 & 4 \\ 4 & 6 & 4 \\ 4 & 4 & 4 \end{bmatrix}} \quad \underset{Z=1}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}}, \tag{4.27}$$

while the lookup table was defined as

$$P_N = [0.00001, 0.01, 0.02, 0.03, 0.05, 0.06, 0.07, 0.1, 0.2, 0.3, 0.5, 0.6, 0.7, 1, 1, \dots]. \tag{4.28}$$

This window function was chosen to favour the assignment of voxels at the same depth as any opaque neighbouring voxels, followed by favouring voxels on the diagonals. This helps to form continuous surfaces, rather than blobs of opaque points.

The visibility terms were also calculated using a lookup table. This was indexed by the number of unassigned subsets that each voxel was a member of. For a five camera setup, this was defined as

$$P_V = [0, 0.001, 0.05, 0.08, 0.1, 1]. \tag{4.29}$$

Both the neighbourhood and visibility lookup tables were derived by brief trial and error, based on what produced the best results. Consequently, the resulting table values are unlikely to be optimal.

To test the effect of these priors on the resulting reconstruction, the greedy algorithm was retested on the synthetic test set using Eq. 4.26. This was performed using a noise variance of $\sigma^2 = 10$ and a robust parameter value of $\lambda_e = 74$. This robust parameter corresponds with an outlier probability of $\lambda_p = 1 \times 10^{-3}$ as given by Eq. 4.13.

As shown in Fig. 4.10(a), the projected square intensity error of the obtained estimate is higher than that obtained without the incorporation of prior information. This is to be expected, since the MAP estimate minimises a combination of the data error and negative log prior probability, rather than simply minimising the data error term alone. Although the projected square error is increased, the resulting scene estimate is

more likely to be similar to the ideal scene. This is observed in the results, with the obtained depth-maps corresponding much more closely with the ideal depth-maps, as shown in Fig. 4.10(b).

Although the square error in the projected intensities is similar to that obtained for the ideal binary opacity voxel parameters in most regions of the scene, there are a number of undesirable artefacts in the reconstruction. First, there is some ghosting of the borders around depth discontinuities in both the intensity and depth-map images, as shown more clearly by the close-ups in Fig. 4.11(b) and (e). There are still a number of errors in the obtained depth-maps, especially around depth discontinuities.

One of the problems with the presented greedy approach is that decisions are made locally. This is important computationally but results in a reconstruction that is unlikely to be globally optimal. This is especially true with the incorporation of smoothness priors, as these are inherently global constraints.

In an attempt to improve the optimisation, an alternative implementation of the smoothness prior is presented. With this approach, the prior probability terms are calculated based on the likelihood that a voxel's neighbours are opaque, rather than on what neighbours have actually been assigned as opaque. This is achieved by convolving the initial benefits, calculated as before, with a smoothing or surface detection filter.

To demonstrate this technique, the initial benefits were calculated using the cost function

$$\text{cost}_j(c, \mathbf{s}) = \frac{1}{2\sigma^2} E_j^o(c, \Omega_j) - \log(V_j(\mathbf{s})). \tag{4.30}$$

This is the same as Eq. 4.26, except the smoothness prior term has been removed from the expression. These initial benefits were then modified using the shaping function

$$\text{benefit}' = \frac{1}{\frac{1}{\text{benefit}} + \kappa}. \tag{4.31}$$

This was necessary to help place greater emphasis on the smaller benefits during the convolution. In these experiments $\kappa$ was set equal to 2. For voxels that had been assigned opaque, a benefit of $1.2/\kappa$ was assigned to help weight neighbouring points in the convolution. The resulting benefits were then convolved with the smoothing windowing function, $W_N$, given in Eq. 4.27.

The resulting square error of the projected intensities and depth-map from the central camera position are shown in Fig. 4.10. Similar results were observed from the other cameras. As can be seen, both the projected intensities and depth-map are not as good as without the filtering. The filtered approach did remove some of the small isolated errors but tended to distort the object boundaries and also remove some of the finer detail.

One of the main problems with the convolution approach to incorporating smoothness priors is that it is not a particularly effective surface filter. As a consequence, the

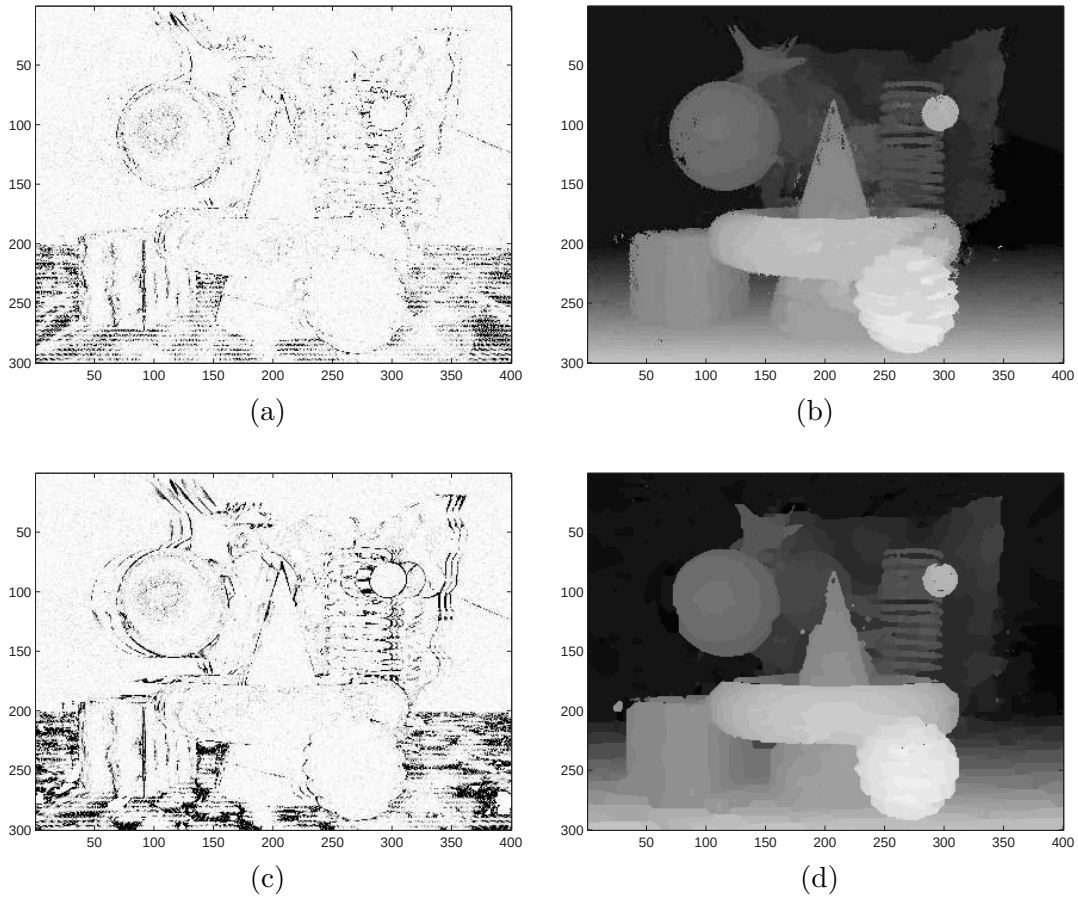


**Figure 4.10** Results of the greedy algorithm with visibility updating and the addition of simple prior terms. (a) The incorporation of prior information results in an increase in the projected square error between the reconstructed central image and original image. This is to be expected, as a combination of both the data error and negative log prior probability must now be minimised. (b) The resulting depth-map from central camera position is significantly improved with the incorporation of prior information. However, there are still numerous large errors, especially near depth discontinuities. Similar results were observed from the other camera positions. (c) Square error between the reconstructed central image and original image, obtained by convolving the voxel benefits with a small window function at each iteration prior to selecting maximum. (d) By filtering the voxel benefits some isolated errors were removed from the resulting depth-map, however object boundaries became distorted and some of the finer detail was lost. The filtered approach is also significantly more computationally expensive.

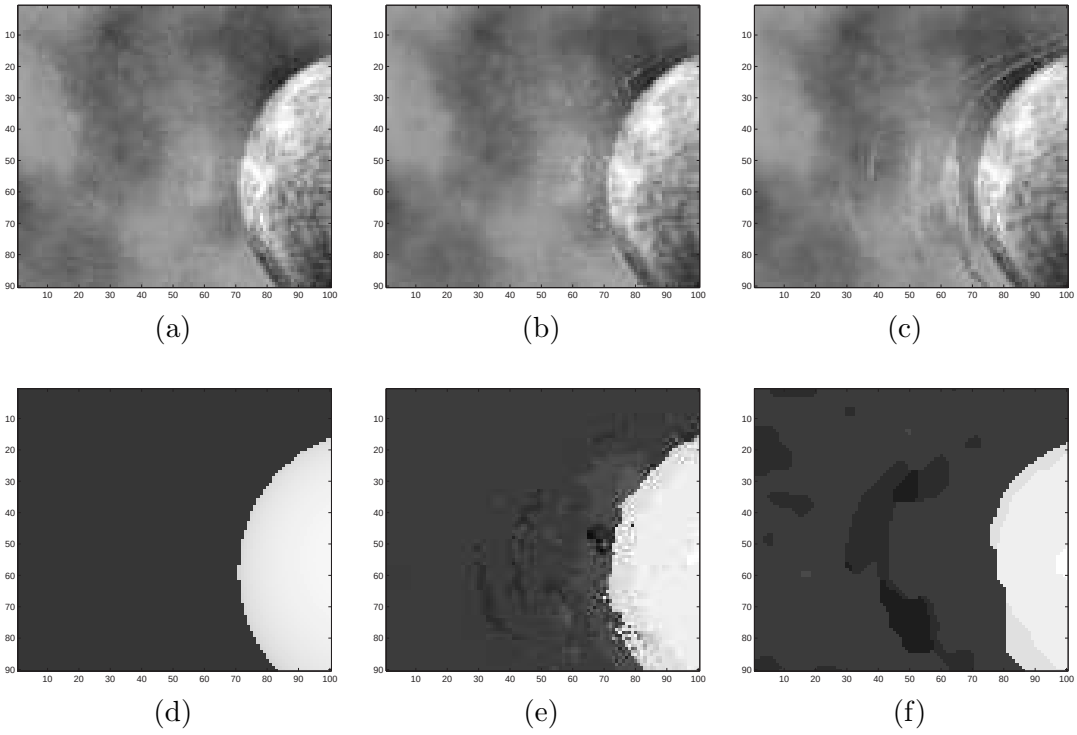


**Figure 4.11** Close up of Fig. 4.10, showing the results of greedy algorithm around the large sphere in the upper left half of the images. (a) The original image intensities from central camera. (b) Ideal depth-map of the actual scene. (b) The projected intensities and (e) depth-map of the greedy algorithm with neighbourhood smoothing. (c) Projected intensities and (f) depth-map of the greedy algorithm with convolution smoothing. In both of the reconstructed images, the background intensities are smoother and correspond more closely with the ideal scene radiances than the original noisy image. However, there is some noticeable ghosting around object boundaries, especially visible in (c).

calculated benefits for each voxel are inaccurate, resulting in an assignment of opaque voxels that is far from optimal. Improved techniques for incorporating smoothness priors and optimising the resulting posterior distribution are discussed in Chapter 5 and Chapter 6.

## 4.7 EFFICIENT GREEDY IMPLEMENTATION

Both the greedy and pixel ray assignment algorithms form an estimate of the scene by iteratively assigning voxels as opaque. This is achieved by assigning a single voxel as opaque at each iteration, until a complete estimate of the scene is formed. For a standard scene model, with dimensions of about $580 \times 300 \times 36$, this will require approximately 200,000 iterations. With such a large number of iterations it is essential that each iteration is implemented as efficiently as possible.

Having assigned a voxel as opaque, the visibility and cost of all affected voxels must be updated. With the greedy algorithm, only those voxels that are members of the same subset, or are neighbours of the assigned voxel, are affected. For a standard sized scene and five camera setup, this equates to only approximately 250 voxels that must

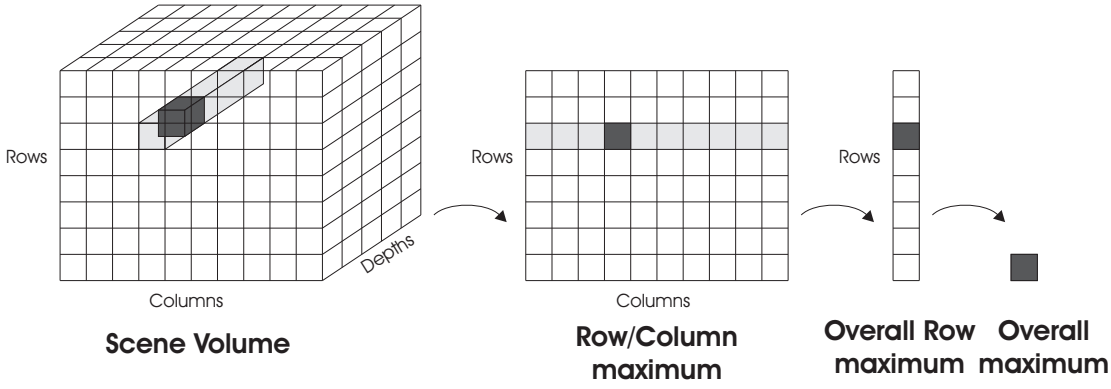


**Figure 4.12** The global maximum within the scene volume can be efficiently updated using a hierarchical approach, where the maximums along each row and plane within the scene are stored, enabling a new maximum to be calculated with minimum computation.

be updated. Therefore, the updating of voxel visibilities and costs can be performed reasonably quickly. Having updated the costs, the benefits of these voxels are then updated accordingly. With the filtered greedy algorithm, where voxel benefits are convolved with a small smoothing function, all benefits within half the dimensions of the smoothing function must also be updated. This significantly increases the number of voxels that need updating and so is noticeable slower than without the convolution.

With the filtered greedy algorithm, the convolution updating can be performed efficiently by multiplying the changes in the unfiltered benefits by the convolution kernel and adding the result to the convolved benefits. This is performed independently for each voxel whose unfiltered benefit has changed.

Having made the appropriate updates, the final step at each iteration is to assign the voxel with maximum benefit as opaque. This requires searching for the maximum benefit within the scene volume. With approximately $580 \times 300 \times 36 \approx 6 \times 10^6$ voxels, this is exceedingly expensive computationally if implemented naively. To improve performance, an efficient technique for updating the maximum benefit is presented.

### 4.7.1 Fast maximisation

To efficiently recompute the maximum benefit at each iteration, a hierarchical approach is used. To begin, the maximum benefit in the $Z$, or depth, direction is calculated for each row and column position within the scene. These maxima are stored in a matrix of row/column maxima. Next, the maximum benefit along each row of this matrix is computed and stored in an third vector of row maxima. Finally, the maximum overall benefit is found by calculating the maximum benefit within the vector of row maxima. To obtain the index of the voxel with maximum benefit, an additional set of index matrices are computed in parallel. These store the indices of the row/column maxima and row maxima. This process is shown diagrammatically in Fig. 4.12.

This requires comparable effort to searching the entire volume for the overall maxi-

mum. The real advantage of this approach comes about when a new maximum must be calculated, following the updating of voxel benefits at each iteration. For each updated benefit, the index and value of the new benefit is first compared with the index and value of the current maximum for that row and column. If the new benefit is greater than the current maximum, then the current row/column maximum is updated with the new benefit. Otherwise, if the new benefit is less than the current maximum, and the indices match, then the current maximum is no longer valid. In this situation, a new maximum must be found along that row/column position. If the new benefit is less than the current maximum, and the indices are different, then the current maximum is valid and no updating is required. This process is then repeated for any row/column maxima which have changed, to update the overall row maximum, and then for any overall row maxima which have changed, to update the global maximum.

## 4.8 DISCUSSION

By incorporating prior information into the greedy reconstruction process, the resulting scene estimate is significantly improved, corresponding more closely with the actual scene, although the re-projected error did increase. This highlights the importance of prior information on the solution, demonstrating that for the scene reconstruction problem, prior information or other regularisation is essential to obtain a reliable estimate of the scene.

Although reasonable results were obtained with the greedy algorithm using simple priors, there are still a number of problems which reduce the quality of the reconstruction. A particular problem that was observed is that the calculated square error associated with each voxel is adversely affected by modelling errors.

As demonstrated in Section 2.3.4, variation in the convolution kernel between images results in the observed voxel radiances changing between images. These changes vary smoothly with differences in viewing angle or position. Un-modelled variations in the observed intensity between images are also caused by errors in the binary opacity assumption. These errors primarily cause difficulties in regions where there is a large variation in intensity between neighbouring voxels, or at object boundaries. To reduce the effect of these errors, a novel projected error measure is presented in Section 6.4.

The optimisation process is likely to get stuck in a local minimum or maximum, since decisions are made locally. The greedy algorithm is also affected by its simplistic incorporation of smoothness priors, as well as its inability to correct poor assignments as the reconstruction progresses. It is also inherently slow, due to its sequential nature, although computation time can be significantly reduced by assigning multiple points at each iteration, as well as efficient updating and maximisation. To overcome some of these problems and attempt to improve the scene estimate, a variety of global optimisation techniques based on belief propagation are presented in Chapter 5 and Chapter 6.

# Chapter 5

## VOLUMETRIC BELIEF PROPAGATION

In the previous chapter an iterative approach for reconstructing the scene was presented. Using a greedy approach, the scene was progressively reconstructed, while updating voxel visibilities at each iteration. This enabled occlusions to be treated correctly, allowing complex scenes, containing many discontinuous surfaces and semi-occluded regions, to be reconstructed. However, the approach makes poor use of prior information, resulting in mediocre reconstructions. It is also inherently sequential and computationally intensive, making it slow to execute.

To overcome these difficulties and improve performance, this chapter presents an alternative approach based on Belief Propagation (BP). Belief propagation is an iterative algorithm that can be used to maximise the joint posterior probability of a set of random variables. Using a Bayesian approach, scene reconstruction can be treated as a statistical inference problem, where the objective is to find the 'best' estimate of the scene given the camera data and any prior information. Such problems arise in a wide variety of disciplines, and have accordingly been well researched. This has lead to the development of numerous, often similar, inference algorithms for solving these problems. Of these, belief propagation has proved to be one of the most efficient and successful [Yedidia et al. 2002b].

Based on local message passing, belief propagation is an efficient technique for finding the most likely state of a probabilistic network given any available evidence. The approach is exact when the network has a tree structure but is only approximate when the network contains cycles. By using an appropriate probabilistic network to model the scene and images, this approach can easily be applied to the scene reconstruction problem. This enables complicated prior models to be used effectively, thereby making better use of any available prior information. It also lends itself to efficient parallel implementations, providing fast computation and the possibility of real time application.

To apply belief propagation to the scene reconstruction problem, the imaging system must first be represented using an appropriate probabilistic model. This model should accurately describe the statistical relationship between the scene parameters and the image data, as well as simplify the interaction between variables, if possible.

By reducing the dependencies within the system, the underlying posterior distribution will be easier to optimise. This often results in a tradeoff between model accuracy and ease of optimisation. Consequently, the choice of model is extremely important, and is an important component of any reconstruction algorithm.

In this chapter belief propagation is applied to a novel volumetric factor graph model of the scene and imaging system. Probabilistic models are introduced in Section 5.1. The max-product and sum-product forms of belief propagation are described in Section 5.2 along with convergence problems. In Section 5.3 a novel volumetric factor graph system model is presented. An efficient procedure for updating the messages and performing belief propagation on the volumetric model is presented in Section 5.4 along with preliminary results showing that the basic belief propagation algorithm is unstable on this network. A novel approach for helping convergence is demonstrated along with results on the synthetic shapes test set.

## 5.1 PROBABILISTIC MODELS

Like many tasks in image and vision computing, the scene reconstruction problem can be expressed as maximising the joint probability distribution over a set of variables. In principle, this problem is trivial and can be solved by searching through all possible combinations to find the one which is most likely. However, even for very small systems such an approach is infeasible, since the number of possible combinations is enormous. Luckily most joint probability distributions contain a considerable amount of structure which can be used to help simplify the problem. Probabilistic models provide a way of using this structure, by explicitly representing the dependencies between variables, usually expressed in graphical form. This enables the joint probability distribution to be compactly represented and lends itself to a number of efficient optimisation procedures.

### 5.1.1 Bayesian networks

Although a variety of probabilistic models exist, perhaps the easiest to understand from an intuitive point of view are Bayesian networks. These are directed acyclic graphs, which explicitly represent the conditional dependencies between system variables. In a Bayesian network, each variable is represented using a single node, with edges between nodes modelling causal impact or probabilistic dependencies between the variables. Associated with each node is a conditional probability distribution $\rho_{X_i|U_i}(x_i|\mathbf{u}_i)$, which defines the likelihood that the node is in state $x_i$, given the state of its parents $\mathbf{u}_i$. The term "parents" is used to denote the subset of nodes which link to a given node.

Using $\mathbf{X} = \{X_1, \ldots, X_N\}$ to denote the system of nodes, the joint probability distribution of the entire system $\rho_X(x)$ is then given by the chain rule [Jensen 2001] for

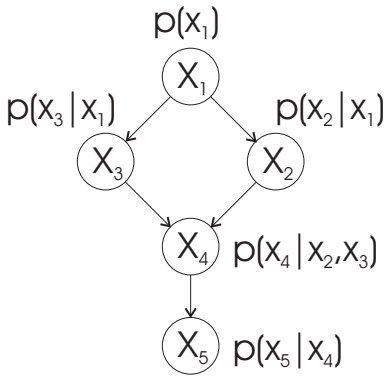


**Figure 5.1** A Bayesian network modelling the environment of a footpath. The five variables $X_1$, $X_2$, $X_3$, $X_4$, $X_5$ represent the season, whether it rained, whether the sprinkler was left on, whether the pavement is wet, and whether the pavement is slippery. These are related to one another by a set of conditional probability distributions, which describe the probability of a node being in a particular state, given the state of its parents.

Bayesian networks, as

$$\rho_X(\mathbf{X}=\mathbf{x}) = \prod_i^N \rho_{X_i|U_i}(x_i|\mathbf{u}_i). \tag{5.1}$$

As an example, consider the rain and sprinkler model presented by Pearl [1996]. This consists of five variables: the seasons of the year ($X_1$), whether it is raining ($X_2$), the state of a sprinkler ($X_3$), whether the pavement is wet ($X_4$), and if it is slippery ($X_5$). These five variables are statistically related to one another.

To represent this system as a graphical Bayesian network, a node is assigned to each variable. Directed links are then added between the nodes to represent direct conditional dependance. Associated with each node is a conditional probability distribution, describing the probability of obtaining each of the node states, given the state of its parents. The resulting Bayesian network is shown in Fig. 5.1. This clearly shows the dependence between the variables and provides a compact representation of the system. The joint probability function $\rho_X(x_1, x_2, x_3, x_4, x_5)$ of the system is given by Eq. 5.1, as

$$\rho(x_1, x_2, x_3, x_4, x_5) = \rho(x_5|x_4)\rho(x_4|x_2, x_3)\rho(x_3|x_1)\rho(x_2|x_1)\rho(x_1), \tag{5.2}$$

where the subscripts of the probability distributions have been dropped for clarity.

The advantage of Bayesian networks is that it is generally straightforward to determine the structure and local conditional probability distributions associated with a particular system. This is particularly true in situations where the system variables have a cause and effect relationship. As a consequence, Bayesian networks are commonly used in a variety of inference problems. They are especially popular in the field of Artificial Intelligence [Pearl 1988, Korb and Nicholson 2004].

One disadvantage of Bayesian networks is their limited ability to represent non causal relationships. They are also unable to represent cyclic dependencies. These

problems can be avoided by using either Markov Random Field or Factor graph models, which are discussed in the next two subsections.

### 5.1.2 Markov random fields

The Markov Random Field (MRF), or Markov network, is an undirected graphical model that represents the dependencies within a full joint probability distribution. This model is especially popular in the image processing domain [Geman and Geman 1984]. The MRF better represents cyclic dependencies than the Bayesian network and is also more intuitive in its representation of non-causal systems.

A Markov Random Field is defined with respect to a neighbourhood system that describes the conditional independencies within the joint probability distribution. Using $\mathbf{N} = \{\mathbf{N}_i : X_i \in X\}$ to represent a neighbourhood system, where $\mathbf{N}_i$ is the set of nodes neighbouring or linking to node $X_i$, a network over $\mathbf{X}$ is said to be a MRF with respect to $\mathbf{N}$, if and only if

$$\begin{aligned} \rho_{X_i}(x_i) > 0, \forall x_i \in \mathbf{x}, \\ \rho_{X_i|\mathbf{X}\backslash X_i}(x_i|\mathbf{x}\backslash x_i) = \rho_{X_i|\mathbf{N}_i}(x_i|\mathbf{n}_i), \forall x_i \in \mathbf{x}, \end{aligned} \tag{5.3}$$

where $\backslash$ is the set minus operator. The set of nodes $\mathbf{N}_i$ are said to form a Markov blanket of a node $X_i$. Given the Markov blanket of $X_i$, every node $X_i$ in a Markov network is conditionally independent of every other node.

The neighbourhood system, $\mathbf{N}$, defines a set of cliques over the network. A clique $k$ is defined as a subset of nodes in $\mathbf{X} = \{X_1, \ldots, X_N\}$ for which every pair of nodes are neighbours, except for single-node cliques. A maximal clique is a clique to which no more nodes can be added. By associating a set of potential functions or clique potentials with each maximal clique, the joint probability distribution modelled by a Markov network is given by

$$P(\mathbf{x}) = \frac{1}{Z} \prod_k \phi_k(\mathbf{x}_k), \tag{5.4}$$

where $\mathbf{x}_k$ is the state of the random variables in the $k^{\text{th}}$ clique, and $Z$ is a normalising constant called a partition function, given by

$$Z = \sum_{x_i \in \mathbf{X}} \prod_k \phi_k(\mathbf{x}_k). \tag{5.5}$$

In many situations the joint probability distribution of the system can be represented using a pairwise MRF, which has a maximum clique size of two. A graphical example of a pairwise MRF model is shown in Fig. 5.2. This provides a simple structure suitable for numerous inference algorithms. These models are commonly used in computer vision and image processing problems, where the objective is to estimate some underlying 2D function or image from a set of observations.

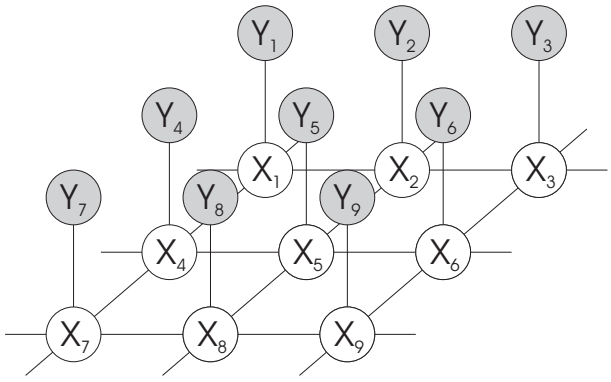


**Figure 5.2** A pairwise Markov Random Field, representing the joint probability distribution of an underlying image $\mathbf{x} = \{x_1, \ldots, x_N\}$ and a set of observations $\mathbf{y} = \{y_1, \ldots, y_N\}$. With a pairwise MRF the maximum clique size is two. Therefore, the joint probability distribution can be represented as a product of quadratic terms, as given by Eq. 5.6.

Modelling the ideal image as an array of hidden or unknown nodes $X_i$, and using $Y_i$ to represent the set of observations, the joint probability distribution for a pairwise MRF is given by

$$\rho_{X,Y}(\mathbf{X}, \mathbf{Y}) = \frac{1}{Z} \prod_{ij} \psi_{ij}(x_i, x_j) \prod_{i} \phi_i(x_i, y_i). \tag{5.6}$$

### 5.1.3 Factor graphs

Factor graphs [Kschischang et al. 2001] are a more recently proposed probabilistic model, becoming increasingly popular in the field of statistical optimisation. These are a generalisation of a Tanner graph [Tanner 1981], where the function nodes can represent any function instead of just parity check constraints.

A factor graph is a bipartite graph that represents the factorisation of a function of several variables. In graph theory, a bipartite graph is a special graph where the set of nodes or vertices can be divided into two disjoint sets such that no links or edges connect any nodes within the same set. With a factor graph, these two sets correspond to a set of variable nodes and a set of factor nodes. The variable nodes $\mathbf{X} = \{X_1, \ldots, X_N\}$ correspond with the system variables, while the factor nodes $\mathbf{Y} = \{Y_A, \ldots, Y_M\}$, represent the local functions $f_A$, $f_B$, $f_C$, $\ldots$, $f_M$, that are used to factorise the joint probability distribution. An edge connects the variable node $X_i$ to the factor node $Y_a$, if and only if $X_i$ is an argument of $f_a$. A simple example of a factor graph is shown in Fig. 5.3.

Using the factor graph model, the joint probability distribution of the system is given by

$$\rho_X(x) = \frac{1}{Z} \prod_{a} f_a(\mathbf{x}_a), \tag{5.7}$$

where $\mathbf{x}_a$ are the arguments to $f_a$ and $Z$ is a normalisation constant.

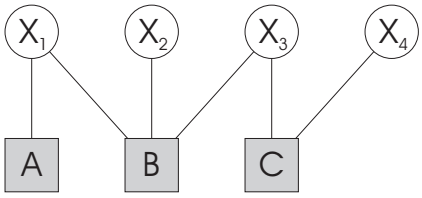


**Figure 5.3** A simple factor graph representing the joint probability distribution $\rho(x_1, x_2, x_3, x_4) = \frac{1}{Z} f_A(x_1) f_B(x_1, x_2, x_3) f_C(x_3, x_4)$, where $Z$ is normalisation constant.

### 5.1.4 Model equivalence

Although the various probabilistic models represent a joint probability distribution differently, it is possible to express any probability distribution using any of these models [Yedidia et al. 2002a]. However, this may require additional nodes to be added into the network, increasing the complexity of the model and making it less intuitive. As a consequence, some models are more suitable than others for representing a particular distribution. Choosing the right model can help clarify the problem and lead to more efficient implementation. In this chapter a factor graph model is used because it is more suitable for representing general distributions, such as the one presented.

## 5.2 BELIEF PROPAGATION

Belief propagation is an iterative inference algorithm based on local message passing that can be used to maximise the marginal or joint posterior probability of a system, at least approximately [Yedidia et al. 2002b]. By modelling the scene and images as a probabilistic model and expressing the scene reconstruction problem as a Maximum A Posteriori (MAP) or Minimum Mean Square Error (MMSE) estimation problem, this algorithm can readily be applied to the scene reconstruction problem.

Proposed by Pearl [1988], belief propagation is one of several closely related message passing algorithms that have been independently developed for solving inference problems on probabilistic models. Equivalent or closely related algorithms include the Viterbi algorithm [Forney 1973], the turbo-decoding algorithm [McEliece et al. 1998], and the Kalman filter [Kalman 1960]. These algorithms are designed to either maximise the joint posterior of the entire system or determine the marginal posterior of individual variables.

Confusingly, there are two forms of the belief propagation algorithm. One form is for maximising the joint posterior of the entire system, while the other is for determining the marginal posterior of individual variables. These are referred to as the *max-product* and *sum-product* algorithms respectively [Pearl 1988]. In both instances, messages are iteratively sent between neighbouring nodes in a graphical probabilistic model until a final solution is obtained.

The algorithms are exact when the graphical model has a tree structure but only approximate when the graph contains cycles or loops. Fortunately, surprisingly good results are often obtained even in the presence of cycles. However, there is no guarantee that the solution will be near optimal, and in some instances the algorithm may not converge to a solution at all. Performance of the algorithm depends on the structure of the network, as well as the local probability distributions associated with each node.

### 5.2.1 Max product algorithm

The max-product algorithm attempts to find the MAP estimate of a system of variables. This algorithm has different forms depending on which network model is being used. Although these are mathematically equivalent, the form of belief propagation for pairwise MRFs is somewhat simpler because it only uses one type of message, and so will be dealt with first.

Beginning with an initial set of beliefs for each node, messages are iteratively passed between neighbouring nodes, updating beliefs until the system converges or some other finishing criterion is met. Using $m_{i\to j}(x_i)$ to represent the message sent from node $i$ to node $j$, the messages for the max product algorithm are given by

$$m_{i\to j}(x_j) := \max_{x_i} \left( \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i) \right), \tag{5.8}$$

where $\psi_{ij}(x_i, x_j)$ and $\phi_i(x_i)$ are the local compatibility functions, or clique potentials of the Markov network, and $N(i) \setminus j$ are the neighbours of node $i$, excluding node $j$.

These messages describe the likelihood that node $j$ is in a particular state given the belief at node $i$. The belief $b_i(x_i)$ at each node is then calculated by multiplying together all the messages coming into that node, giving

$$b_i(x_i) = \kappa\phi_i(x_i) \prod_{j\in N(i)} m_{j\to i}(x_i), \tag{5.9}$$

where $\kappa$ is a normalisation constant. Since the messages can be multiplied by a constant without affecting the beliefs, the message update can alternatively be expressed as

$$m_{i\to j}(x_j) := \max_{x_i} \left( \psi_{ij}(x_i, x_j) \frac{b_i(x_i)}{m_{j\to i}(x_i)} \right). \tag{5.10}$$

In most situations, the belief propagation algorithm will converge to the same solution independent of the starting conditions, if it is going to converge at all. Therefore, the choice of initial messages is somewhat arbitrary, as it should not affect the resulting reconstruction. In this work, as with most belief propagation implementations, messages are initialised to a uniform distribution.

Having applied a fixed number of iterations, or run the belief propagation algorithm

to convergence, the most likely scene estimate is then found by selecting the state of each node with the maximum belief. If several nodes have more than one state with maximum belief, then the states of these nodes must be chosen to be consistent with each other. This problem can usually be avoided by adding a small varying random offset to the compatibility functions $\phi_i(x_i)$, to prevent multiple states having the same belief.

So far the max-product algorithm has been described for operating on pairwise MRFs. An equivalent algorithm can be derived for factor graphs. For operating on factor graphs, there are two types of messages; one from the factor nodes to the variable nodes, and the other from the variable nodes to the factor nodes. Using $n_{i\to a}(x_i)$ to denote the message from variable node $i$ to factor node $a$, and $m_{a\to i}(x_i)$ to denote the message from factor node $a$ to variable node $i$, the message update rules for factor graphs are given by

$$n_{i\to a}(x_i) := \prod_{b\in N(i)\setminus a} m_{b\to i}(x_i), \tag{5.11}$$

and

$$m_{a\to i}(x_i) := \max_{\mathbf{x}_a\setminus x_i} f_a(\mathbf{x}_a) \prod_{j\in N(a)\setminus i} n_{j\to a}(x_j), \tag{5.12}$$

where $N(i)\setminus a$ denotes all the nodes that are neighbours of node $i$, and $\max_{\mathbf{x}_a\setminus x_i}$ denotes the maximum over all variables $\mathbf{x}_a$ that are arguments of $f_a$ except for $x_i$. As with pairwise MRFs, the beliefs for each variable node are obtained by taking the product of all incoming messages, giving

$$b_i(x_i) = \kappa \prod_{a\in N(i)} m_{a\to i}(x_i). \tag{5.13}$$

For a standard depth-map representation with dimensions $300 \times 580$ and 36 depths with 4 neighbour connectivity, this results in approximately $25\times10^6$ messages. Assuming parallel implementation, an entire set of both old and new messages must be stored at each iteration. Using double precision 8-byte numbers this corresponds to 400MB of memory plus any additional overhead.

### 5.2.2 Sum product algorithm

Instead of calculating the maximum of the joint probability distribution, a common problem is to determine the marginal probability distribution for each variable. This is useful, since the mean of the marginal probabilities gives the Minimum Mean Square Error (MMSE) estimate of the system.

To do this, belief propagation can be applied using the sum-product algorithm. This is similar to the max-product algorithm, except that a summation is used in the message updating instead of a maximisation. The message updates for the sum-product

algorithm on pairwise MRFs are given by

$$m_{i\to j}(x_j) := \sum_{x_i} \phi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i). \tag{5.14}$$

For operating on a factor graph, the message updates are

$$n_{i\to a}(x_i) := \prod_{b\in N(i)\setminus a} m_{b\to i}(x_i), \tag{5.15}$$

and

$$m_{a\to i}(x_i) := \sum_{\mathbf{x}_a\setminus x_i} f_a(\mathbf{x}_a) \prod_{j\in N(a)\setminus i} n_{j\to a}(x_j). \tag{5.16}$$

The calculation of belief is the same as for the max-product algorithm.

### 5.2.3 Convergence and accuracy of belief propagation

As shown by Pearl [1988], belief propagation is exact when the graphical model has a tree structure but only approximate when the graph contains cycles or loops. In the presence of loops, belief propagation tends to produce good results providing it converges [Murphy et al. 1999, Frey and MacKay 1998, Heskes 2003]. This idea was qualified by Weiss and Freeman [2001] for the max-product algorithm, who showed that the fixed points of the max-product algorithm correspond to a strong local minimum, within a large neighbourhood region.

However, on certain networks belief propagation may fail to converge. As noted by Heskes [2003], there appears to be two causes of non-convergence. The first is caused by too large a step size, similar to what can occur in gradient decent minimisation. This is easily avoided by damping the message updates. Heskes [2003] performed this in the logarithmic domain. In this thesis, an approach similar to Weiss [Murphy et al. 1999] is used, where the new messages are a weighted addition of the previous and updated messages, giving

$$m_{i\to j}(x_j)^{(t)} := (1-\mu)m_{i\to j}(x_j)^{(t)} + \mu m_{i\to j}(x_j)^{(t-1)}, \tag{5.17}$$

where $\mu$ is the weighting term, referred to as the momentum.

The other cause of non-convergence is the inherent local instability of the fixed points of the belief propagation algorithm. As shown by Yedidia et al. [2002a] and Yedidia et al. [2002b], the belief propagation algorithm can only converge to a fixed point that is also a stationary point of the Bethe approximation to the free energy of the joint probability distribution. This idea was extended by Heskes [2003], who showed that constrained minimisation of the Bethe free energy can be turned into an unconstrained saddle-point problem. Heskes [2003] also proved that belief propagation with message damping has the same local stability properties as a gradient decent-

ascent procedure. Therefore, depending on the local curvature at the saddle point, belief propagation will either converge or not converge.

In networks where the fixed points of the belief propagation algorithm are unstable, the solution is to use a more complex algorithm that is guaranteed to converge, such as the double loop algorithm [Heskes 2006, Heskes 2003] or the CCCP algorithm [Yuille 2002]. These algorithms work by iteratively minimising an upper bound on the function, which is constructed to ensure the algorithm converges. This is usually achieved by fixing some of the node states to make the bounding function convex. The convex function is then solved using belief propagation and the resulting minimum used to calculate a new upper bound. The process is repeated until a local minimum is obtained.

In addition to problems with convergence, the Bethe approximation to the free energy may in some cases be rather poor. Improved results can often be obtained by using a better approximation, such as the Kikuchi cluster variation method [Kikuchi 1951] or junction graph method of Aji and Mceliece [2001]. Both of these are special cases of the more general region graph method, for which generalised belief propagation algorithms have been developed [Yedidia et al. 2002b, Yedidia et al. 2002a]. Alternative approaches for improving the results of the sum-product algorithm include reducing the absolute value of the log-likelihood ratio of the messages [Yazdani et al. 2004] and using the more complex cavity method [Mooij et al. 2007].

### 5.2.4 Implementation of belief propagation

One of the main benefits of belief propagation is that it particularly suitable for parallel implementation, since all message updates can be performed synchronously. This enables extremely fast computation on custom hardware with parallel architecture. Unfortunately when implementing on a PC, calculations must be performed sequentially. This gives rise to a number of different message update schedules [Tappen and Freeman 2003]. One option is to perform a synchronous update schedule, where each node first computes the messages to be sent to its neighbours. Once every node has computed the messages, the messages are passed to each node and used to compute the next round of messages. This simulates a parallel implementation, but requires a large amount of memory, since all the messages must be stored at each iteration. A synchronous update schedule is used in this thesis.

An alternative update schedule is to sequentially update the messages to each node, using the new messages immediately for any subsequent calculations. With a grid of nodes this is often done by propagating the messages in one direction first and then repeating the process in the other directions. One advantage of this method is that information is quickly propagated across the scene. For a synchronous update schedule on a network of width $W$, it takes $W$ iterations for information from one side of the

network to reach the other. Using a sequential schedule only one iteration is required to propagate this information, significantly speeding up the convergence rate. Another approach for a grid network, is to divide the nodes into a bipartite graph and then only propagate messages from one half of the graph at each iteration. As proposed by Felzenszwalb and Huttenlocher [2004], this approach halves the number of messages that must be computed, as well as halving the memory requirements.

Improvements to the belief propagation algorithm can also be made by efficient calculation of the message updates. This is achieved by using the underlying structure of the neighbourhood compatibility functions to reduce the computation. Such an approach is used in Section 5.4 to feasibly compute the messages passed from factor nodes $I_j$ to scene nodes $S_i$ in the volumetric model. A similar approach is used by Felzenszwalb and Huttenlocher [2004] for computing the messages in several simple pairwise models. Felzenszwalb and Huttenlocher [2004] also present an hierarchical approach, to improve the convergence speed of belief propagation.

## 5.3 VOLUMETRIC FACTOR GRAPH MODEL

To apply belief propagation to the scene reconstruction problem, the scene and image formation process must be modelled using an appropriate probabilistic network. In this section a volumetric approach is presented where a full 3D discrete model of the system is represented using a factor graph model. The use of a factor graph model enables the imaging process and prior information to be represented in a compact and intuitive way, which is not possible with either a pair-wise MRF or Bayesian model.

With this approach, the scene is modelled as a 3D array of variables that represent the bandlimited opacity and radiance at discrete points within the scene. The statistical interaction between the scene variables and the image data for a discrete volumetric model, as given by Theorem 3, Chapter 2, is shown graphically in Fig. 5.4. In this diagram, scene variables and pixel data are represented using spherical nodes, with black lines indicating direct statistical interactions. The resulting model is equivalent to a Markov model of the system.

Adopting a Bayesian approach, similar to that presented in Section 4.1, the MAP estimate of the scene can be expressed as,

$$S_{\text{MAP}}(\mathbf{c}) = \arg\max_{\mathbf{s}} \left[\rho_{C|S}(\mathbf{c}|\mathbf{s})\rho_S(\mathbf{s})\right], \tag{5.18}$$

where $\mathbf{c} = \{c_1, c_2, \ldots, c_N\}$ represents the observed camera data corresponding with the set of image pixels, $\mathbf{C} = \{C_1, C_2, \ldots, C_N\}$, and $\mathbf{s} = \{s_1, s_2, \ldots, s_M\}$ represents an estimate of the scene parameters $\mathbf{S} = \{S_1, S_2, \ldots, S_M\}$. Modelling the prior probability distribution as a pairwise function and assuming the observed pixel values are conditionally independent given the scene parameters, the MAP estimate can be expanded

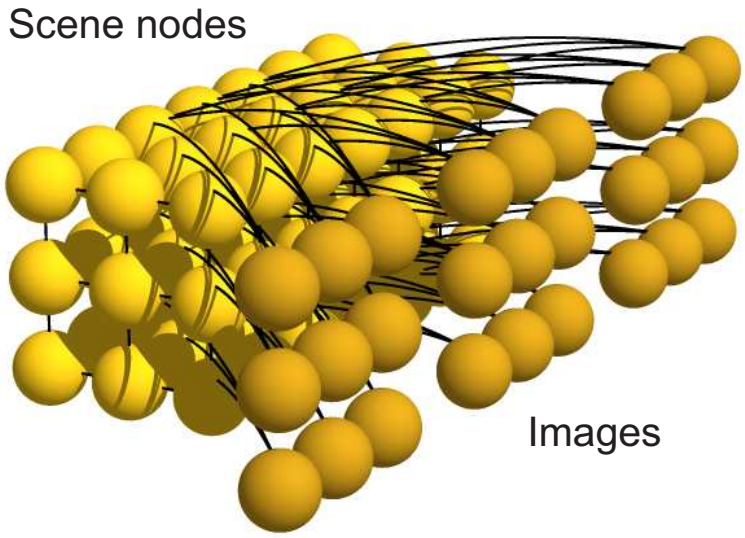


**Figure 5.4** Volumetric voxel based model of scene and imaging system. Scene variables and pixel data are represented using spherical nodes, with black lines indicating direct statistical interactions. The resulting model is equivalent to a Markov model of the system.

to give

$$S_{\text{MAP}}(\mathbf{c}) = \arg\max_{\mathbf{s}} \left[ \prod_{i=1}^{N} \rho_{C_i|S}(c_i|\mathbf{s}) \prod_{j\,k} f_{jk}(s_j, s_k) \right], \tag{5.19}$$

where $f_{jk}(s_j, s_k)$ is the compatibility function between neighbouring scene variables describing the prior probabilities. Using $\mathbf{s}_i$ to denote the states of voxels $\mathbf{S}_i$ that directly affect the observed intensity at pixel $i$, this can more conveniently be written as

$$S_{\text{MAP}}(\mathbf{c}) = \arg\max_{\mathbf{s}} \left[ \prod_{i=1}^{N} \rho_{C_i|S}(c_i|\mathbf{s}_i) \prod_{j\,k} f_{jk}(s_j, s_k) \right]. \tag{5.20}$$

To simplify the interactions between variables and reduce the elements in $\mathbf{S}_i$, an approach similar to that presented in Section 4.1 is used, where the data probabilities are defined over a set of perturbed pixel positions, $\acute{\mathbf{C}}_i(\mathbf{s}_i)$, where $\mathbf{s}_i$ are the states of voxels $\mathbf{S}_i$ located along the extended pixel ray $i$. The index $(\mathbf{s}_i)$ indicates that the perturbed pixel position, and hence intensity, corresponds to the projected position of the nearest opaque voxel within the set $\mathbf{S}_i$. This allows the data probability terms to be expressed as

$$\rho_{C_i|S}(c_i|\mathbf{s}_i) \equiv \rho_{\acute{C}_i|S}(\acute{c}_i(\mathbf{s}_i)|\mathbf{s}_i). \tag{5.21}$$

By introducing the term $f_i(\mathbf{s}_i) = \rho_{\acute{C}_i|S}(\acute{c}_i(\mathbf{s}_i)|\mathbf{s}_i)$ to denote the data compatibility function of the variables $\mathbf{s}_i$, the MAP estimate can be expressed as

$$S_{\text{MAP}}(\mathbf{c}) = \arg\max_{\mathbf{s}} \left[ \prod_{i=1}^{N} f_i(\mathbf{s}_i) \prod_{j\,k} f_{jk}(s_j, s_k) \right]. \tag{5.22}$$

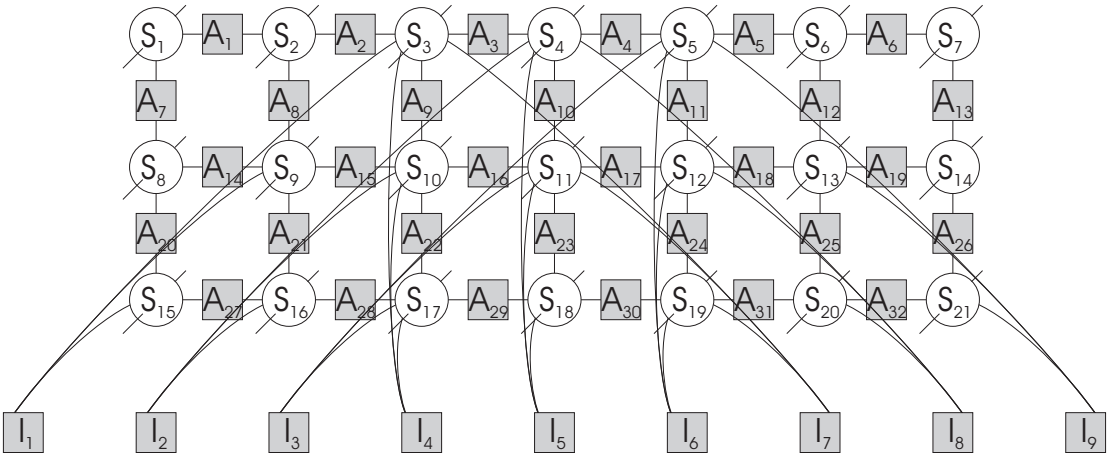


**Figure 5.5** Volumetric factor graph model of scene and imaging system. In this diagram only a 2D horizontal slice of the scene and images is shown for clarity. The darker square nodes represent the factor nodes, while the light circular nodes represent the scene parameters at discrete points within the scene volume. Factor nodes $I_i$ represent the probability of obtaining the data given scene parameters $S_i$, while factor nodes $A_j$, represent the prior compatibility function between neighbouring scene parameters. Because the observed data is a fixed known quantity it is incorporated into the factor functions, rather than explicitly represented as an additional set of variables.

This expresses the MAP estimate as a maximum over the joint probability distribution of the system, described in terms of its factors. This distribution has the same form as the factor graph model given in Eq. 5.7, except it has been broken down into two different types of factors: the factors $f_{jk}(s_j, s_k)$, corresponding to the prior distribution, and the factors $f_i(\mathbf{s}_i)$, corresponding to the probability of obtaining the image data.

The resulting factor graph of the system is shown graphically in Fig. 5.5. In this model, the factor nodes $A_j$ correspond with the factor functions $f_{jk}(s_j, s_k)$, while the factor nodes $I_i$ correspond with the factor functions $f_i(\mathbf{s}_i)$. This model highlights the underlying structure of the joint probability distribution and provides a platform for statistical optimisation algorithms such as belief propagation.

In addition to representing the overall structure of the joint probability distribution, the individual factor functions $f_i(\mathbf{s}_i)$ and $f_{jk}(s_j, s_k)$ must be defined and represented. As with the overall probability distribution, these factor functions often contain a degree of structure which can be taken advantage of by the optimisation algorithm.

### 5.3.1 Data factor functions

The data factor functions $f_i(\mathbf{s}_i)$ describe the conditional probability of obtaining pixel intensities $\acute{c}_i(\mathbf{s}_i)$ given the state $\mathbf{s}_i$ of the scene nodes lying along the $i^{\text{th}}$ extended pixel ray. The perturbed pixel intensities $\acute{c}_i(\mathbf{s}_i)$ correspond to the pixel intensity at the projected image position of the nearest opaque voxel within $\mathbf{S}_i$, and are found by interpolating the image data. As with the greedy algorithm presented in Section 4.5, linear interpolation is used in these experiments.

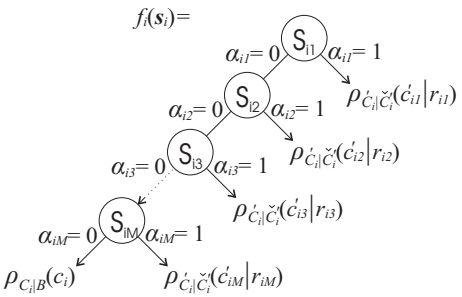


**Figure 5.6** Tree structure representing $f_i(\mathbf{s}_i)$. The conditional probabilities are grouped depending on the opacity of the scene nodes.

Assuming robust gaussian noise and binary scene opacities, the conditional probability of obtaining the perturbed pixel intensities is given by Eq. 4.10 in Section 4.1, as

$$\begin{aligned}\rho_{\acute{C}_i|S}(\acute{c}_i(\mathbf{s}_i)|\mathbf{s}_i) &= \rho_{\acute{C}_i|\check{C}_i^r}(\acute{c}_i(\mathbf{s}_i)|r_{\zeta_i(\mathbf{s}_i)}(\theta_i)) \\ &= \max\left(\frac{1}{\sigma\sqrt{2\pi}}\exp\frac{-(\acute{c}_i(\mathbf{s}_i)-r_{\zeta_i(\mathbf{s}_i)}(\theta_i))^2}{2\sigma^2}, \lambda_p\right), \end{aligned} \tag{5.23}$$

where $\zeta_i(\mathbf{s}_i)$ is the index of the nearest opaque voxel along the $i^{\text{th}}$ extended pixel ray and $r_{\zeta_i(\mathbf{s}_i)}(\theta_i)$ is the radiance of that voxel in the direction of the $i^{\text{th}}$ sensor element. If there is no opaque voxel anywhere along the extended pixel ray, then $\rho_{\acute{C}_i|S}(\acute{c}_i(\mathbf{s}_i)|\mathbf{s}_i) = \rho_{C_i|B}(c_i)$. The term $\rho_{C_i|B}(c_i)$ describes the probability of obtaining $c_i$ given the prior distribution of the background radiance. If nothing is known about the background, as will be assumed in this work, then $\rho_{C_i|B}(c_i)$ can be represented using a uniform distribution.

By representing the state of each voxel, $s_i = \{\alpha_i, r_i\}$ by its opacity $\alpha_i$ and radiance $r_i$, and using $s_{ij}$ to denote the $j^{\text{th}}$ voxel in $\mathbf{s}_i$, the data factor functions can be compactly represented using the tree structure shown in Fig. 5.6. The probability functions $\rho_{\acute{C}_i|\check{C}_i^r}(\acute{c}_{ij}|r_{ij})$ in the tree structure are given from Eq. 5.23 as

$$\rho_{\acute{C}_i|\check{C}_i^r}(\acute{c}_{ij}|r_{ij}) = \max\left(\frac{1}{\sigma\sqrt{2\pi}}\exp\frac{-(\acute{c}_{ij}-r_{ij})^2}{2\sigma^2}, \lambda_p\right), \tag{5.24}$$

where $\acute{c}_{ij}$ indicates the perturbed pixel intensity corresponding to the projected image position of voxel $s_{ij}$, and $r_{ij}$ is the radiance of voxel $s_{ij}$. To clarify the tree structure representation, consider evaluating the data factor function $f_i(\mathbf{s}_i)$, given by

$$f_i(\mathbf{s}_i) = f_i(\alpha_{i1}, r_{i1}, \alpha_{i2}, r_{i2}, \alpha_{i3}, r_{i3}) = f_i(0, 35, 1, 62, 1, 86), \tag{5.25}$$

where the perturbed pixel intensities are given by $\acute{\mathbf{c}}_i = \{\acute{c}_{i1}, \acute{c}_{i2}, \acute{c}_{i3}\} = \{54, 59, 63\}$. Using the tree structure, the data factor function is given by $f_i(\mathbf{s}_i) = \rho_{\acute{C}_i|\check{C}_i^r}(\acute{c}_{i2}|r_{i2}) = \rho_{\acute{C}_i|\check{C}_i^r}(59|62)$, since $\alpha_{i1} = 0$ and $\alpha_{i2} = 1$.

### 5.3.2 Prior factor functions

The prior factor functions $f_{jk}(s_j, s_k)$ describe the prior probability distribution of the scene variables. For the volumetric factor graph model presented in this chapter, these are pairwise functions, describing the local compatibility between neighbouring scene nodes. Higher order factor functions can be used and are necessary to represent certain prior distributions accurately. However, this will affect the structure of the factor graph and increase its complexity. In these experiments only pairwise prior distributions are considered.

One of the most commonly used priors is that opacities and radiances within the scene are likely to be correlated between neighbouring voxels. This can be expressed in terms of pairwise factor functions, where a higher weighting or probability is given to neighbouring voxels that are in the same or similar state. Using $s_i = \{\alpha_i, r_i\}$ to describe the state of each voxel in terms of its opacity $\alpha_i$ and radiance $r_i$, this prior can be incorporated by defining the prior factor functions as

$$f_{jk}(s_j, s_k) = f^R_{jk}(r_j, r_k|\alpha_j, \alpha_k) f^O_{jk}(\alpha_j, \alpha_k) \rho_j(\alpha_j) \rho_k(\alpha_k), \tag{5.26}$$

where $f^R_{jk}(r_j, r_k|\alpha_j, \alpha_k)$ is a function representing the compatibility between neighbouring radiances, given the voxel opacities, $f^O_{jk}(\alpha_j, \alpha_k)$ is a function representing the compatibility between neighbouring opacities, and $\rho_j(\alpha_j)$ is the probability distribution of opacities for voxel $j$.

Assuming binary opacities and that the radiance of all transparent regions is zero, the compatibility function $f^R_{jk}(r_j, r_k|\alpha_j, \alpha_k)$ can be defined as

$$f^R_{jk}(r_j, r_k|\alpha_j, \alpha_k) = \begin{cases} \frac{1}{Z_1} f_R(r_j - r_k) & \alpha_j = 1, \alpha_k = 1 \\ \frac{1}{Z_2} & \alpha_j = 0, \alpha_k = 1, r_j = 0 \\ \frac{1}{Z_2} & \alpha_j = 1, \alpha_k = 0, r_k = 0 \\ 1 & \alpha_j = 0, \alpha_k = 0, r_j = 0, r_k = 0 \\ 0 & \text{otherwise,} \end{cases} \tag{5.27}$$

where $f_R(r_j - r_k)$ is a function describing the compatibility between radiances of two opaque voxels, and $Z_1$ and $Z_2$ are normalisation constants to ensure each term integrates or sums to one over the range of neighbouring scene radiances. In this work $f_R(r_j - r_k)$ was defined as the triangular shaped function

$$f_R(r_j - r_k) = \beta_R - \frac{\beta_R - 1}{L} |r_j - r_k|, \tag{5.28}$$

where $\beta_R$ is a shaping function that specifies the ratio between the peak and the minimum of the distribution, and $L$ is the range of scene radiances.

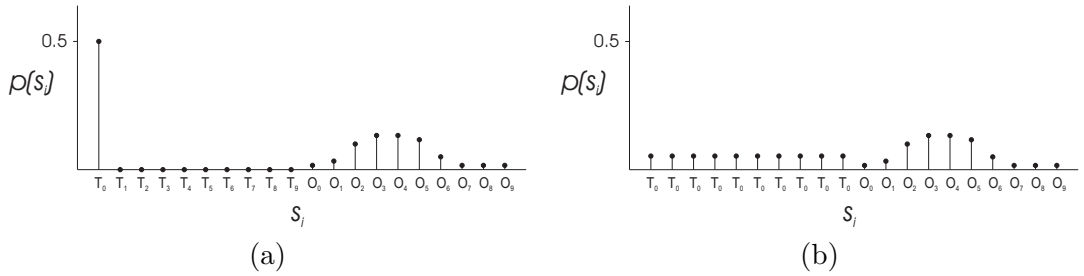


**Figure 5.7** (a) Posterior probability distribution of a scene variable, with transparent states $T_0$ to $T_9$ and opaque states $O_0$ to $O_9$, corresponding with radiances $r_i = 0$ to 9. In this example, because the summation over opaque states is larger than the summation over transparent states, the scene variable is more likely to be opaque than transparent. However, the MAP estimate will simply select the single most likely state, corresponding with state $T_0$. (b) By assuming all transparent states are identical with a radiance of zero, the MAP estimate will instead correspond with $O_5$, since this becomes the singly most likely state.

The opacity compatibility function $f^O_{jk}(\alpha_j, \alpha_k)$ is defined as

$$f^O_{jk}(\alpha_j, \alpha_k) = \begin{pmatrix} \frac{\beta_O}{2} & \frac{1-\beta_O}{2} \\ \frac{1-\beta_O}{2} & \frac{\beta_O}{2} \end{pmatrix}, \tag{5.29}$$

where $\beta_O$ represents the compatibility between voxels in the same opacity state.

The problem with defining the prior factor function in this way is that assuming the prior probability of each voxel being opaque is 0.5, the MAP estimate will favour voxels to be in a transparent state with radiance equal to zero (Fig. 5.7(a)). The reason for this is that although there are numerous possible opaque states, corresponding to different voxel radiances, there is only one possible transparent state. Therefore, even though the overall probabilities of being opaque or transparent are equal, the transparent state with radiance equal to zero will have the greatest individual probability.

This highlights one of the main problems of using the MAP estimate. Although the MAP estimate is the most likely estimate of the scene, it is not necessarily likely to be close to the actual scene. This is especially true when dealing with mixed distributions that contain both continuous and discrete variables.

To avoid this problem, one could first obtain the MAP estimate over the scene opacities independent of radiance and then optimise over the scene radiances. Unfortunately, it is not clear how this could be performed. In this work a simpler approach is taken, where it is assumed that all transparent states have a radiance of zero (Fig. 5.7(b)). The effect of this is to scale down the transparent compatibility terms in Eq. 5.27, since the four sub-terms corresponding to different neighbouring opacity combinations must each sum to one.

### 5.3.3 Discrete variables

To apply the standard belief propagation algorithm, the system variables must be discrete. For optimising continuous systems, there are similar alternative algorithms, such as expectation minimisation [Minka 2001b] and generalised belief propagation [Heskes and Zoeter 2003]. However, these are slightly more complex and rely on the probability distributions being reasonably smooth or having some underlying structure.

Assuming binary transmittances within the scene, the opacity of each voxel can be represented by a discrete binary variable. This has the value zero if the voxel is transparent and one if the voxel is opaque. Voxel radiances, on the other hand, have a wide continuous distribution. Therefore, in general, a large number of discrete states are needed to represent these accurately. For most grey scale images 256 states are used. This requires a large amount of memory and results in a system model that is extremely slow to optimise.

To reduce the number of discrete states required to represent voxel radiances, basic information about the possible radiances for each voxel is obtained from the image data. Assuming a voxel is visible in one or more of the images, its radiance is likely to correspond reasonably closely with one of the observed pixel intensities of that point. In situations where a voxel is not visible in any of the images, its radiance is impossible to estimate reliably without strong priors, and since it will not affect the observed data, it is of little consequence to the reconstruction. Therefore, the discrete states representing a voxel's radiance are chosen to correspond with the observed pixel intensities of that voxel in each of the images.

## 5.4 EFFICIENT VOLUMETRIC BELIEF PROPAGATION

Given the factor graph model of the scene, the max-product belief propagation algorithm can be applied to try and find the MAP estimate of the scene. For a factor graph model, the update messages for the max-product algorithm are given by Eq. 5.11 and Eq. 5.12. These can be broken down into two different forms: the first, for messages sent to and from the data factors and the other, for messages sent to and from the prior factors.

Using the expression for a variable's belief given in Eq. 5.13, the message updates for the volumetric factor graph system model are given by

$$n_{S_i \to A_{ij}}(s_i) := \frac{b_{S_i}(s_i)}{m_{A_{ij} \to S_i}(s_i)}, \tag{5.30}$$

$$n_{S_i \to I_j}(s_i) := \frac{b_{S_i}(s_i)}{m_{I_j \to S_i}(s_i)}, \tag{5.31}$$

$$m_{A_{ij} \to S_i}(s_i) := \max_{s_j} f_{ij}(s_i, s_j) n_{S_j \to A_{ij}}(s_j), \tag{5.32}$$

$$m_{I_j \to S_i}(s_i) := \max_{\mathbf{s}_j \setminus s_i} f_i(\mathbf{s}_j) \prod_{S_k \in \mathbf{S}_j \setminus S_i} n_{S_k \to I_j}(s_k), \tag{5.33}$$

where

$$b_{S_i}(s_i) = \kappa \prod_{I_k \in N_{Ii}} m_{I_k \to S_i}(s_i) \prod_{A_{ij} \in N_{Ai}} m_{A_{ij} \to S_i}(s_i), \tag{5.34}$$

$\mathbf{S}_j$ are the neighbours of factor node $I_j$, $N_{Ii}$ and $N_{Ai}$ are the data node and prior node neighbours of $S_i$ respectively and $\kappa$ is a normalisation coefficient. The variables $\mathbf{S}_j$ correspond to the voxels lying along the $j^{\text{th}}$ extended pixel ray.

The first two of these messages, corresponding to the outgoing messages from variable node $S_i$, are easy to compute and involve a simple division. The third message, from prior node $A_{ij}$ to variable $S_i$, is also reasonably easy to compute. This requires $N^2$ multiplications and a maximisation over $N^2$ states, where $N$ is the number of states of each variable. This can be reduced further, if only the prior compatibility between opacities is used.

The final set of messages are rather difficult to compute, as they require $N^M$ multiplications and a maximisation over $N^M$ states, where $M$ is the number of elements in $\mathbf{S}_i$. For a standard sized scene model of dimensions $300 \times 580 \times 36$, $\mathbf{S}_i$ will contain 36 elements. Assuming five cameras, this gives $5^{36} \approx 10^{25}$ multiplications and a maximisation over an equal number of states for the computation of a single message.

To reduce the number of computations, the structure of the data compatibility functions shown in Fig. 5.6, can be used to compute the message updates more efficiently. Using a hierarchical approach, the maximisation can be broken down into a number of smaller maximisations over each node in the tree structure. Depending on the opacity of a voxel $i$, there are two different expressions for the resulting updates. These are given by

$$m_{I_j \to S_i}(s_i \neq t) := \max \left[ a_{j1i}, a_{j2i}, \ldots, a_{j(i-1)i}, a_{jii} \right], \tag{5.35}$$

$$m_{I_j \to S_i}(s_i = t) := \max \left[ a_{j1i}, a_{j2i}, \ldots, a_{j(i-1)i}, a_{j(i+1)i}, \ldots, a_{jMi}, a_{jBi} \right], \tag{5.36}$$

where $a_{jki}$ are the sub-maxima over the $k^{\text{th}}$ node in the tree, and $s_i \neq t$ is used to indicate all states except the transparent state, $t$. The terms $a_{j1i}$ to $a_{jBi}$ are given by

$$\underset{(k \neq i)}{a_{jki}} = \max_{s_{jk} \neq t} \max_{\substack{s_{jn} \in \mathbf{s}_j \\ n>k \\ n \neq i}} \left[ \rho_{C_i|\acute{C}_i}(c_i|s_{jk}) n_{S_{jk} \to I_j}(s_{jk} \neq t) \prod_{\substack{m<k \\ m \neq i}} n_{S_{jm} \to I_j}(s_{jm} = t) \prod_{\substack{n>k \\ n \neq i}} n_{S_{jn} \to I_j}(s_{jn}) \right], \tag{5.37}$$

$$a_{jii} = \max_{\substack{s_{jn} \in \mathbf{s}_j \\ n>i}} \left[ \rho_{C_i|\acute{C}_i}(c_i|s_{ji}) \prod_{m<i} n_{S_{jm} \to I_j}(s_{jm} = t) \prod_{n>i} n_{S_{jn} \to I_j}(s_{jn}) \right], \tag{5.38}$$

$$a_{jBi} = \rho_{C_i|B}(c_i) \prod_{m \neq i} n_{S_{jm} \to I_j}(s_{jm} = t). \tag{5.39}$$

By making use of the independencies within each term, these can equivalently be written

as

$$a_{\substack{jki \\ (k\neq i)}} = \max_{s_{jk}\neq t}\left[\rho_{C_i|\acute{C}_i}(c_i|s_{jk})n_{S_{jk}\rightarrow I_j}(s_{jk}\neq t)\right]\prod_{\substack{m<k \\ m\neq i}} n_{S_{jm}\rightarrow I_j}(s_{jm}=t)\prod_{\substack{n>k \\ n\neq i}}\max_{s_{jn}}(n_{S_{jn}\rightarrow I_j}(s_{jn})), \tag{5.40}$$

$$a_{jii} = \rho_{C_i|\acute{C}_i}(c_i|s_{ji})\prod_{m<i} n_{S_{jm}\rightarrow I_j}(s_{jm}=t)\prod_{n>i}\max_{s_{jn}}(n_{S_{jn}\rightarrow I_j}(s_{jn})), \tag{5.41}$$

$$a_{jBi} = \rho_{C_i|B}(c_i)\prod_{m\neq i} n_{S_{jm}\rightarrow I_j}(s_{jm}=t). \tag{5.42}$$

To compute these terms efficiently, common components within the terms can be calculated initially and then used to simplify the computation of each term, thus avoiding repeating unnecessary calculations. Using this approach, the maximisation terms are given by

$$a_{\substack{jki \\ (k<i)}} = \frac{d_{jk}e_{jk}f_{jk}}{\max_{s_{ji}}(n_{S_{ji}\rightarrow I_j}(s_{ji}))}, \tag{5.43}$$

$$a_{\substack{jki \\ (k>i)}} = \frac{d_{jk}e_{jk}f_{jk}}{n_{S_{ji}\rightarrow I_j}(s_{ji}=t)}, \tag{5.44}$$

$$a_{jii} = \rho_{C_i|\acute{C}_i}(c_i|s_{ji})e_{jk}f_{jk}, \tag{5.45}$$

$$a_{jBi} = \frac{\rho_{C_i|B}(c_i)e_{jk}}{n_{S_{ji}\rightarrow I_j}(s_{ji}=t)}, \tag{5.46}$$

where

$$d_{jk} = \max_{s_{jk}\neq t}\left[\rho_{C_i|\acute{C}_i}(c_i|s_{jk})n_{S_{jk}\rightarrow I_j}(s_{jk}\neq t)\right], \tag{5.47}$$

$$e_{jk} = \prod_{m<k} n_{S_{jm}\rightarrow I_j}(s_{jm}=t), \tag{5.48}$$

$$f_{jk} = \prod_{n>k}\max_{s_{jn}}(n_{S_{jn}\rightarrow I_j}(s_{jn})). \tag{5.49}$$

### 5.4.1 Results with uniform prior

To test the effectiveness of the max-product belief propagation algorithm for maximising the joint posterior probability distribution of the scene, the algorithm was tested on the synthetic shapes test set, shown in Section 4.3.1. Because of the large amount of memory required to store all the messages and local compatibility functions in the belief propagation algorithm, the algorithm was only tested on a subset of the rows from each image. The following results show the reconstruction for a horizontal slice through the scene at row one hundred.

To help convergence, the updated messages were weighted by the previous messages, as given by Eq. 5.17. This was done using a momentum value of $\mu = 0.2$. A small random

offset was also added to the prior probability of each node, to help prevent equal belief from occurring. To model the image noise, the data factor functions were calculated using a noise variance of $\sigma = 10$ and a robustness term of $\lambda_p = 0.01$.

First, the max-product algorithm was tested assuming a uniform prior distribution. This was achieved by setting the prior factor functions equal to a constant. The resulting system was found to be unstable, even if considerable momentum was added to the message updates. Results obtained after 200 and 250 iterations are shown in Fig. 5.9(b) and (c). For comparison, the visible surfaces of the ideal voxel parameters are shown in Fig. 5.9(a).

With this system, the calculated MAP estimate, obtained by taking the maximum belief at each iteration, tended to cycle inwards and outwards with an approximate period of 40 iterations. This corresponded roughly with the range of depths within the scene. As discussed in Section 5.2.3, instabilities in the belief propagation algorithm can be avoided by using a more complex double loop algorithm. However, this adds additional complexity to the problem and is slower to converge. Non-convergence of the belief propagation algorithm is also an indication that the resulting optimisation from the double loop algorithms is likely to be a poor approximation to the true map estimate.

### 5.4.2 Belief Accentuation

To help convergence of the belief propagation algorithm, a novel approach is presented, where the beliefs are accentuated at each iteration. This helps to polarise the beliefs and related messages, forcing the algorithm to converge to a solution around the local operating point. Although no guarantee is given about the accuracy of this solution, results indicate that a reasonable solution is obtained in most cases.

To implement this technique, the beliefs at each node were modified by applying a sigmoidal shaped function. This was achieved by raising the probability ratio of each state to a power slightly greater than one. Using $b'_i$ to denote the modified belief at node $i$, the resulting belief modifications are given by

$$b'_i = \frac{r'_i}{1 + r'_i}, \tag{5.50}$$

where

$$r'_i = \left( \frac{b_i}{1 - b_i} \right)^{\gamma}, \tag{5.51}$$

and $\gamma$ is the shaping term that determines the amount of accentuation. If $\gamma$ is less than one, the effect is to dampen the beliefs. For $\gamma = 2$ the resulting function is shown in Fig. 5.8.

On networks where belief propagation had already converged, accentuating the beliefs had no effect on the solution. However, if the algorithm had not yet converged,

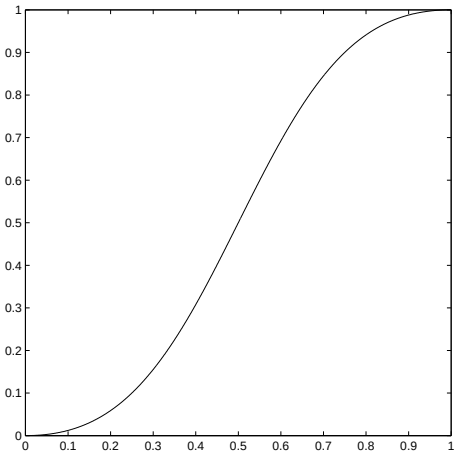


**Figure 5.8** Sigmoidal curve showing the effect of using Eq. 5.50 to accentuated the beliefs. This helps to polarise the beliefs and improve convergence of the max-product algorithm.

accentuating the beliefs often caused the algorithm to converge to a different solution. This solution was invariably a less optimal solution of the MAP estimate. To prevent this from occurring, and help ensure the algorithm converged to a good solution in non-convergent cases, the accentuation was only applied after a fixed number of iterations. The accentuation was then gradually increased until the solution converged. Numerical problems were encountered if the beliefs were accentuated too heavily and became equal to zero, preventing convergence in some cases.

To demonstrate this approach, the beliefs at each node were modified after 120 iterations using Eq. 5.50. Beginning with $\gamma = 1$, the value of $\gamma$ was increased by 5% at each iteration until $\gamma$ was greater than 10. This occurred after another 48 iterations. At this point the algorithm was terminated since convergence had usually been achieved. By applying this approach to the previous system, the resulting reconstruction is shown in Fig. 5.9(d).

By comparing the resulting reconstruction with that obtained from the greedy algorithm, shown in Fig. 5.9(e), using no prior information, it is apparent that the scene structure is similar in both. The diffuse sponge like nature of the reconstruction is a result of trying to minimise the data error term alone. By increasing the number of semi-occluded regions, the scene radiances can be chosen to correspond more closely with the small subset of images which observe each point. The sum of absolute differences between the reconstructed and original images was around 1,900 for both algorithms, however, the sum of square errors was higher for the belief propagation algorithm. This indicates that the belief propagation has fewer but larger errors.

To obtain a fair comparison with the greedy algorithm, the max-product algorithm was retested with the robust parameter $\lambda_p$ set equal to zero, since this was used for testing the greedy algorithm without prior information. Results were generally improved, with the belief propagation algorithm typically giving a slightly lower sum of absolute

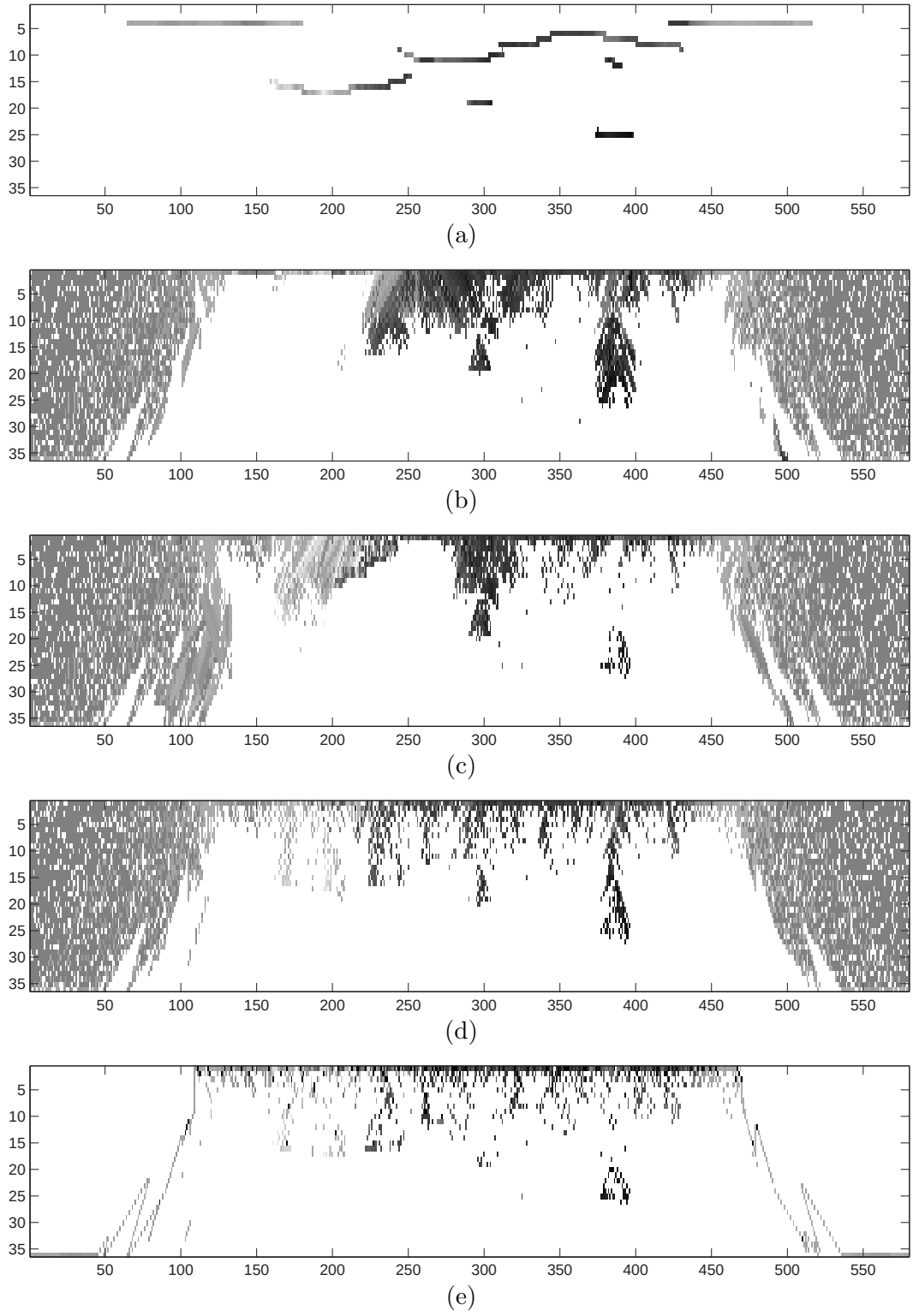


**Figure 5.9** Results from the max-product algorithm on a horizontal slice through the synthetic shapes test set at row 100. (a) Ideal binary opacity scene reconstruction. (b) Results from the volumetric belief propagation approach without prior information after 200 iterations. (c) Results after 250 iterations, showing that the max-product algorithm has not yet converged. (d) By accentuating the beliefs using Eq. 5.50, the max-product algorithm was made to converge to a stable solution. (e) Results obtained from the greedy algorithm using no prior information.

differences than the greedy algorithm. However, the sum of square errors remained significantly higher at around 15,000, compared with 5,900 for the greedy algorithm. With no prior information and $\lambda_p = 0$, both algorithms try to minimise the mean square error. Therefore, in the absence of any prior information, the greedy algorithm appears to produce the best results.

### 5.4.3 Results with smoothing priors

To test the effect of incorporating prior information into the reconstruction, the max-product algorithm was re-run using an opaque compatibility term of $\beta_O = 0.9$ and a radiance compatibility term of $\beta_R = 2$. As was the case without prior information, the max-product algorithm failed to converge without accentuating the beliefs. By applying accentuation after 120 iterations and increasing $\gamma$ by 5% at each subsequent iteration, the resulting reconstruction is shown in Fig. 5.10(a).

As shown, the reconstruction is significantly improved, with voxels grouping themselves into cohesive regions of both opacity and radiance. However, the solution still tends to favour sunken or concave surfaces, where only a small subset of the cameras observe most regions. By increasing the opacity compatibility term further, to $\beta_O = 0.999$, the correlation between opacities is increased, resulting in a reconstruction that is closer to the ideal scene, as shown in Fig. 5.10(b).

Although the results are improved, the obtained reconstruction is still not particularly close to the ideal scene. There are still two sunken surfaces and the three smaller foreground surfaces in the ideal scene have been connected to the background surface. Comparing the original image intensities shown in Fig. 5.10(c) with the projected image intensities, shown in Fig. 5.10(d), it is apparent that the obtained reconstruction matches the image data reasonably closely. However, there are a few small differences which are noticeable. The sum of absolute differences in intensity for both the reconstructed scene, using $\beta_O = 0.999$, and the ideal binary voxel parameters were both around 5,900. The sum of square errors was slightly higher for the reconstructed scene, at around 122,000, compared with the ideal model, which gave 53,961.

The key difference in the projected intensities between the reconstructed scene and the ideal scene was not the overall error but the location of these errors. This is highlighted by comparing the projected square errors for both scenes, as shown in Fig. 5.10(e) and (f). With the ideal binary voxel parameters, the majority of the voxel errors corresponded to regions where there is a large change in intensity between neighbouring voxels, or with depth discontinuities. With the max-product reconstruction, the projected errors did not have such a correspondence with the scene radiances and were often grouped together, making them much more visually noticeable.

The other observation was that $\beta_O$ was required to be close to one to ensure the scene was reasonably smooth and similar to the ideal scene. By plotting a histogram of

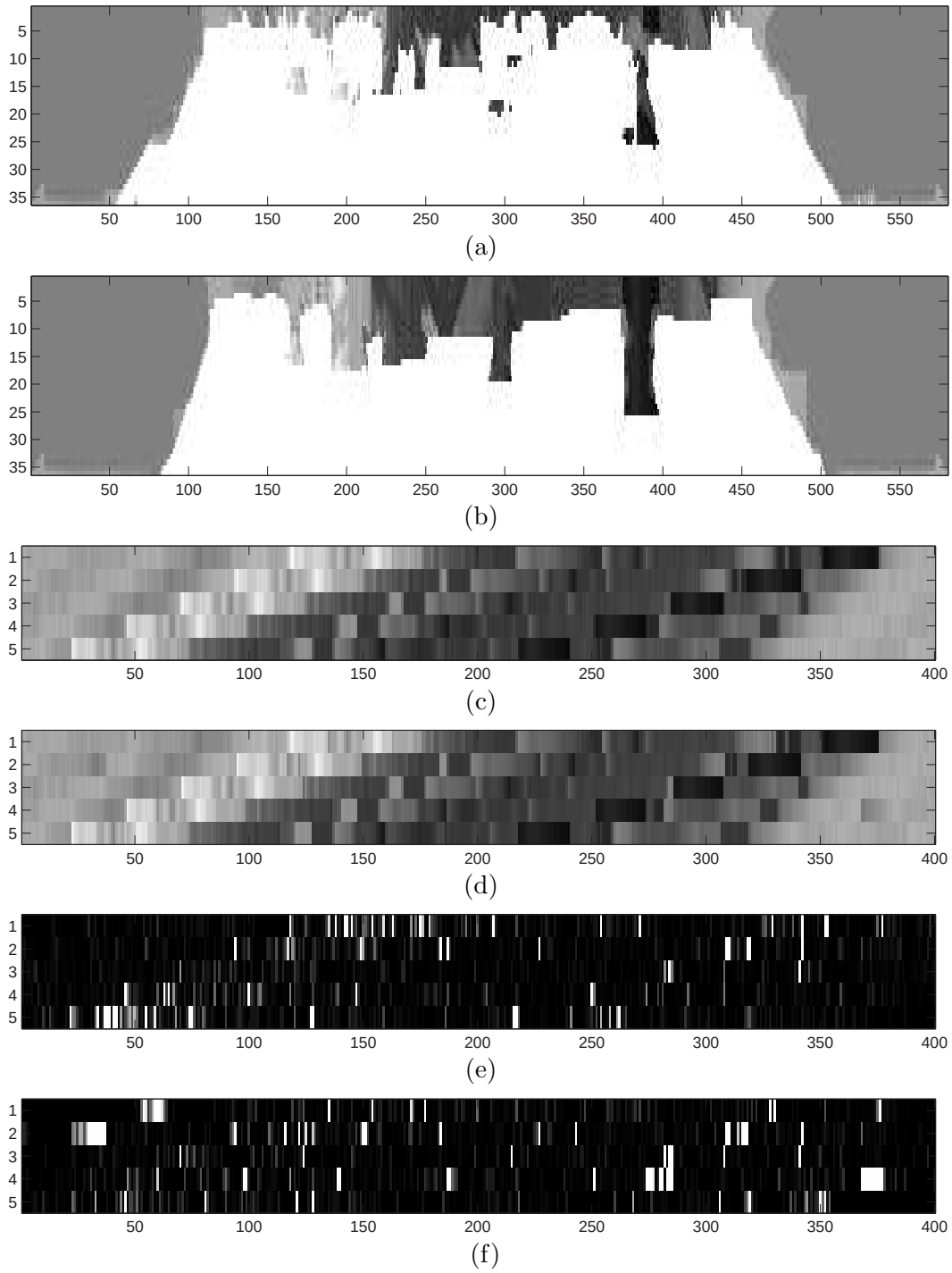


**Figure 5.10** Results from the max-product algorithm with the inclusion of prior information. (a) Results obtained using an opaque compatibility term of $\beta_O = 0.9$. (b) By increasing the prior opacity term to $\beta_O = 0.999$ the scene reconstruction becomes smoother and closer to the actual scene. (c) Original image intensities along row 100 of the five input images. (d) Projected image intensities corresponding with reconstructed scene using $\beta_O = 0.999$. (e) Square error in projected intensities resulting from ideal binary opacity model. These errors are a result of the limitations of the binary opacity approximation. (f) Square error in projected intensities from reconstructed scene, highlighting the difference in the distribution of the errors from the ideal model.

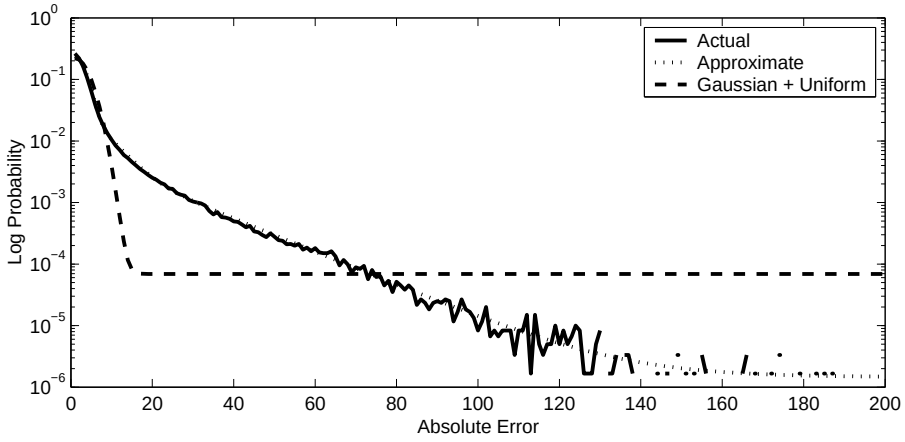


**Figure 5.11** Comparison of the modelled noise distribution using a robust Gaussian with actual noise associated with a binary opacity model. A more complex noise model is also shown, however in this instance, improving the noise model did not seem to effect the results significantly.

the projected data errors for the ideal binary voxel parameters, as shown in Fig. 5.11, it is observed that the modelled noise distribution, given by Eq. 5.24, does not correspond particularly closely with the actual system noise.

To test the effect that improving the data functions has on the resulting reconstruction, the conditional probability terms given by Eq. 5.24, were replaced with a more complex function that corresponded more closely with the observed noise distribution. This is plotted in Fig. 5.11.

The resulting reconstruction is shown in Fig. 5.12(b). To help compare results between algorithms only the visible surfaces are displayed. For this test, an opacity compatibility term of $\beta_O = 0.95$ was used to try and approximate the percentage of neighbouring voxel pairs in the ideal scene parameters which had the same opacity. As observed, modifying the data factor function terms in this instance had little effect on the reconstruction. This can partly be attributed to the fact that both distributions have a similar width for the main peak and both allow some outliers with small probability.

The effect of increasing the noise variance term was also tested, by re-running the algorithm using the original noise model with a variance of $\sigma = 100$ and a prior opacity compatibility term of $\beta_O = 0.99$. The resulting scene estimate is shown in Fig. 5.12(c). For comparison, the results from the two variations of the greedy algorithm using prior information are shown in Fig. 5.12(d) and (e).

The resulting projected square error for the max-product reconstruction using the accurate noise model was 84,900, while the projected square error for the original noise model with a variance of $\sigma = 100$ was 137,236. This was comparable to the greedy algorithm, which had a projected square error of 59,270 for the simple neighbourhood smoothing and 184,599 for the more complex convolved smoothing. In both cases, the projected square error increased as more weighting was placed on the prior information.

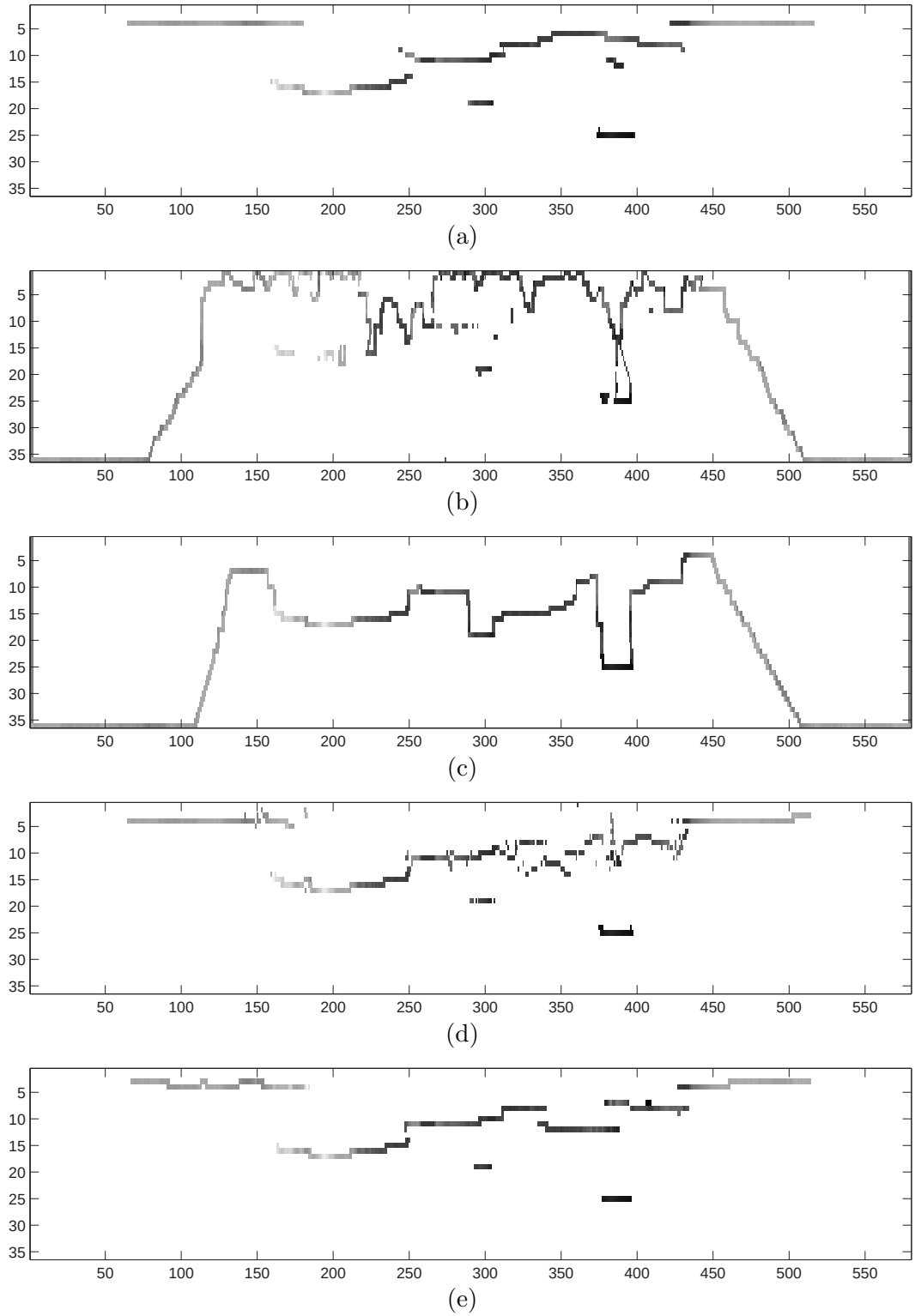


**Figure 5.12** Results from the volumetric belief propagation algorithm are compared with those obtained from the greedy algorithm. (a) Ideal scene reconstruction of visible surfaces (b) Results from belief propagation algorithm using accurate noise model and an opacity compatibility term of $\beta_O = 0.95$. (c) Results from belief propagation algorithm using original noise model with an increased variance of $\sigma = 100$ and prior opacity compatibility term of $\beta_O = 0.99$. (d) Results from greedy algorithm using simple neighbourhood smoothing term. (e) Results from greedy algorithm using the convolution approach for implementing surface smoothing .

Because of the different priors that are used by the greedy and max-product algorithms, it is hard to compare the performance of the two at optimising the joint probability of the system. However, for a given projected square error in intensity, the obtained visible surfaces were similar in their smoothness and continuity.

Despite these similarities, the obtained reconstructions were very different between the two algorithms. On this test set, the greedy algorithm tended to produce an estimate of the scene which corresponded more closely with the ideal voxel parameters. This was quantified by comparing the square errors in the resulting depth-maps between the two algorithms. Because of the large errors in the depth estimate around the edges of the scene for the max-product algorithm, only the inner region of the scene was compared. For the max-product algorithm, the resulting square errors in the depth-map were 58,767 for the accurate noise model and 23,643 for the original noise model with a variance of $\sigma = 100$. The square errors for the greedy algorithm were 8,671 and 12,136 with neighbourhood smoothing and convolved smoothing respectively.

A possible reason for the improved results with the greedy algorithm is the use of the visibility prior, which is not used in the max-product algorithm. This prior helps to favour the reconstruction of surfaces which are visible in a large number of images. Another difference is in the application of the smoothing priors. With the greedy algorithm, smoothness priors are applied to scene surfaces, favouring the reconstruction of continuous surfaces with limited variations in depth, while with the volumetric belief propagation approach, prior information is applied to favour the reconstruction of continuous volumes rather than surfaces.

Although the volumetric belief propagation approach produces a likely estimate of the scene that fits the observed data and the prior information, the approach is very memory intensive and computationally slow. This is a result of using the full volumetric model to represent both the scene opacities and radiance at a large number of discrete points within the scene. To improve convergence, and reduce the computational requirements, an alternative approach based on dynamic belief propagation is presented in the next chapter.

# Chapter 6

# DYNAMIC BELIEF PROPAGATION

In the previous chapter a volumetric approach to the reconstruction problem was presented, where a full volumetric model of the scene and imaging system was optimised using the max-product belief propagation algorithm. This gave promising results, and allowed prior information and visibility interactions to be modelled and optimised using a unified approach. However, the resulting algorithm was slow to converge and extremely memory intensive. The network was also found to be unstable, requiring the beliefs at each node to be accentuated to force the solution to converge.

To help improve convergence and reduce the computational requirements, this chapter presents an alternative approach based on dynamic belief propagation. Unlike standard belief propagation, the proposed dynamic belief propagation algorithm modifies the local probability distributions from iteration to iteration so as to better model the overall system. This allows the complex joint probability distribution of the scene and imaging system to be accurately approximated using a simple probabilistic model. This model is easier to optimise, while maintaining the visibility interactions between voxels.

A simple depth-map model of the scene and imaging system, similar to that presented by Sun et al. [2002], is described in Section 6.1. An alternative volumetric model is presented in Section 6.2. Updating of the factor functions is introduced in Section 6.3. This allows the visibility interaction between voxels to be incorporated into the model. Finally in Section 6.4 an alternative approach for calculating the reliability of a voxel estimate is described.

## 6.1 DEPTH-MAP MRF MODEL

One of the simplest approaches to modelling the scene is to represent the scene as a discrete depth-map. With this approach the scene is represented using a 2D array of variables that describe the depth of the nearest opaque surface along a given line of sight. In most situations the projected depth-map positions are chosen to correspond with the pixel locations in one of the images. This image is referred to as the reference image.

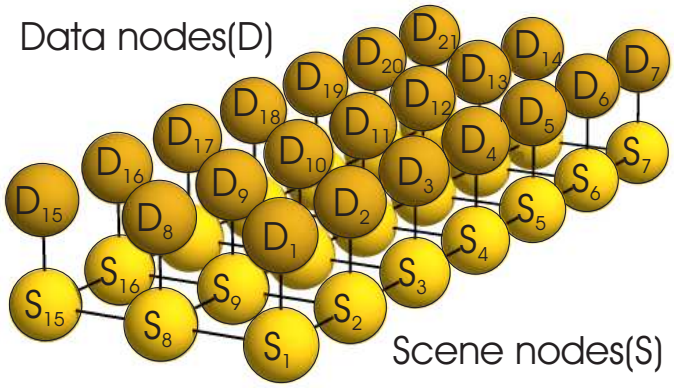


**Figure 6.1** Depth-map model showing the relationship of the scene nodes to the data nodes. Scene variables, $S$, and data terms, $D$, are represented using spherical nodes, with black lines indicating statistical pair-wise interactions.

Assuming the visibilities are known, the joint probability distribution of the system can be obtained to within a scale factor by taking the exponential of Eq. 4.15, given in Section 4.1. This enables the conditional probability of obtaining the data to be expressed as the product of the independent terms corresponding to the visible opaque voxels in the estimate. By evaluating these terms for each $x, y$ position as a function of surface depth, and ignoring any invisible voxels in the reference image, the conditional probability distribution can be modelled as a product of independent functions associated with each variable in the depth-map model.

The prior probability distribution of the depth-map model can also be approximated as a product of local factor functions between neighbouring scene variables. This allows the overall joint probability distribution of the system to be represented using a pairwise MRF, as shown in Fig. 6.1. In this model the local conditional probability distributions associated with the observed data are expressed in terms of a pairwise function between the scene variables and a set of "observation" nodes.

This approach has recently been applied to the scene reconstruction problem with reasonable success [Sun et al. 2003, Boykov et al. 2001]. In the work of Boykov et al. [2001] graph cuts are used to optimise the energy function associated with the joint probability distribution, while Sun et al. [2003] applied the max-product belief propagation algorithm.

A problem with both of these approaches is that the visibility interaction between voxels is modelled poorly. They also suffer from one of the key problems of using a single depth-map model, which is that the resulting scene estimate is only complete from the point of view of the reference camera.

By ignoring voxels occluded in the reference image but visible in one or more of the other images, these algorithms attempt to maximise an approximate posterior distribution rather than the true posterior. In some situations these two distributions will vary

significantly from one another. Consequently, the resulting reconstruction may be far from the global optimum. To overcome this problem a novel known-visibility volumetric model is presented in the following section.

## 6.2 VOLUMETRIC KNOWN-VISIBILITY MODEL

To improve the system model, a known-visibility model of the system is presented. This model ensures the scene is complete with respect to all the camera images, while avoiding having to model the scene radiances or visibility interactions between points. To achieve this, the scene is modelled using a 3D array of voxels or variables whose binary states correspond to the opacities within the scene. The conditional probability of obtaining the data is then modelled using the assumption that the visibility of each opaque voxel is known. As with the depth-map model, the known-visibility assumption allows the conditional probability distribution to be expressed as a product of independent terms corresponding with each of the scene variables.

The simplification of the conditional probability distribution through the known-visibility assumption relies on the property that the scene estimate is complete with respect to all the camera images. If this is not true, then additional background terms must be included, as given by Eq. 4.4. With the single depth-map model, a complete estimate of the scene is ensured, at least with respect to the reference image, since the surface is defined at each $x, y$ position. However, this is not the case with a volumetric model.

To ensure the volumetric model is complete with respect to all the camera images, an additional set of factor functions must be included in the model. These functions are associated with the sets of voxels along each pixel ray, and have a probability of one if there is at least one opaque voxel along the pixel ray and a probability of zero otherwise. In situations where the scene may not be complete, these factor functions can be modified to reflect the probability of obtaining the observed data given the background statistics.

By approximating the prior distribution as a set of pairwise interactions, the resulting joint probability distribution of the system can be represented using a factor graph model, as shown in Fig. 6.2. This has a similar structure to the full volumetric factor graph model shown in Fig. 5.5 but with additional known-visibility data factors and pixel-ray completeness factors replacing the data factors in the full visibility mode.

Although these two models appear similar, the factor functions are very different. With the full visibility model described in Section 5.3, the data factors describe the probability of obtaining the pixel data given the opacity and radiance of all voxels lying along that pixel ray. However, in the known-visibility model the pixel ray factor functions describe the probability that the scene estimate is complete along each pixel ray.

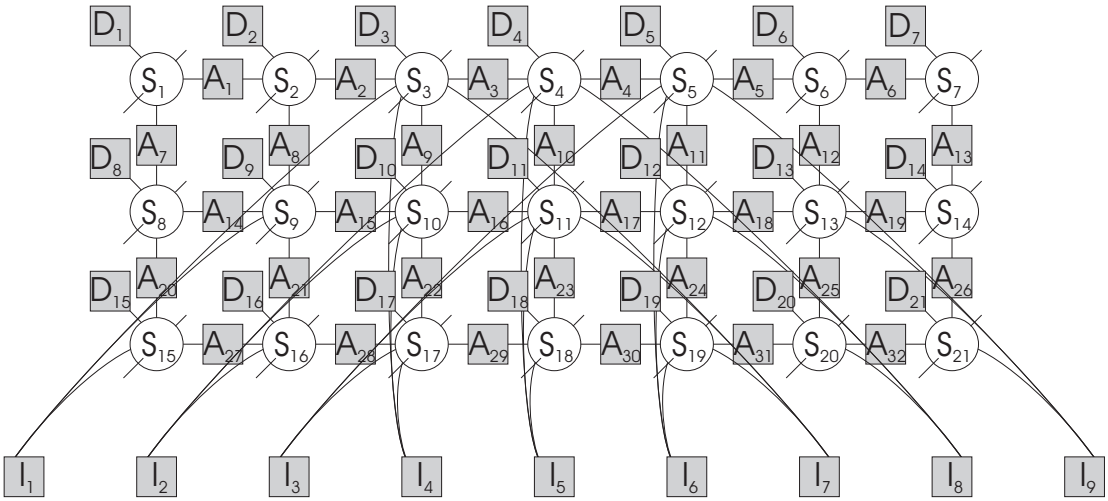


**Figure 6.2** Volumetric known-visibility factor graph model.

### 6.2.1 Efficient calculation of messages

As with the data factors in the full visibility model presented in Section 5.3, the message updates from the pixel ray factor functions to the neighbouring scene variables involve a multiplication followed by a maximisation over all the possible states of the neighbouring scene variables. In this case, the direct computation of message updates is slightly simpler, since each scene variable has only two possible states, corresponding with binary scene opacities, rather than numerous states required to represent both the opacity and possible scene radiances. However, with a typical scene of approximately 36 discrete depths this still results in $2^{36} \approx 10^{11}$ different combinations, which is impractical to compute.

To considerably reduce the computational requirements, the local probability structure can again be used to simplify the message updating. From Eq. 5.12, the message updates $m_{I_j \to S_i}(s_i)$ from the pixel ray factor $I_j$, to the neighbouring scene variable $S_i$, for the max-product algorithm, are given by

$$m_{I_j \to S_i}(s_i) := \max_{\mathbf{s}_j \setminus s_i} f_i(\mathbf{s}_j) \prod_{S_k \in \mathbf{S}_j \setminus S_i} n_{S_k \to I_j}(s_k), \tag{6.1}$$

where $\mathbf{S}_j$ is the set of scene nodes along the $j$th pixel ray, $n_{S_k \to I_j}(s_k)$ is the message from scene node $S_k$ to factor node $I_j$, and $f_i(\mathbf{s}_j)$ is the pixel ray factor function. For a given combination of voxel states along a pixel ray, the factor function will equal one if there is at least one opaque voxel along the pixel ray and zero otherwise. As with the data factors in the full volumetric model, the pixel ray completeness factors can be represented using a tree structure, as shown in Fig. 6.3.

Using the tree structure, and the property that the messages should sum to one, the computation of message updates can be expressed as a maximisation over a number

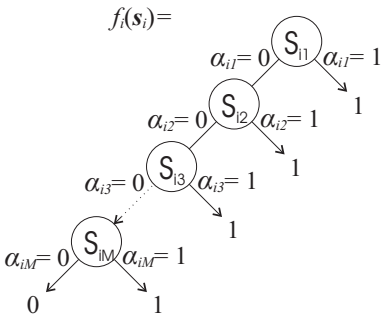


**Figure 6.3** Tree structure representing the pixel ray factor function in the known-visibility factor graph system model.

of sub-maxima, giving

$$m_{I_j \to S_i}(0) := \begin{cases} \displaystyle\prod_{S_k \in \mathbf{S}_j \setminus S_i} \max\left(n_{S_k \to I_j}(s_k)\right) & \exists k, \text{ s.t. } n_{S_k \to I_j}(1) > n_{S_k \to I_j}(0), \\ \displaystyle\max\left(\frac{n_{S_k \to I_j}(1)}{n_{S_k \to I_j}(0)}\right) \prod_{S_k \in \mathbf{S}_j \setminus S_i} \max\left(n_{S_k \to I_j}(s_k)\right) & \text{otherwise,} \end{cases} \tag{6.2}$$

$$m_{I_j \to S_i}(1) := \prod_{S_k \in \mathbf{S}_j \setminus S_i} \max\left(n_{S_k \to I_j}(s_k)\right), \tag{6.3}$$

where 0 and 1 are used to denote transparent and opaque states respectively. These equations can be re-arranged to give

$$m_{I_j \to S_i}(0) := \kappa \min\left(1, \max\left(\frac{n_{S_k \to I_j}(1)}{n_{S_k \to I_j}(0)}\right)\right), \tag{6.4}$$

$$m_{I_j \to S_i}(1) := \kappa, \tag{6.5}$$

where $\kappa$ is a normalisation constant to ensure the messages sum to one. The computation of the prior factor message updates is the same as for the full-visibility model except only binary states are considered.

The message updates from the known-visibility data factor nodes to the corresponding scene variable are given by taking the exponent of Eq. 4.19 for both binary states and then multiplying by a normalisation constant to ensure the messages sum to one.

### 6.2.2 Results

To evaluate the performance of the known-visibility volumetric model for performing scene reconstruction using the max-product belief propagation algorithm, the known-visibility model was tested on the synthetic shapes test set. To reduce computational requirements the algorithm was tested on a horizontal slice through the scene at row one hundred. This also allowed the results to be compared directly with those of the full visibility model.

To begin, the model was tested using a uniform prior probability distribution. The results from this are shown in Fig. 6.4(b). As observed the results are very similar to those of the assignment algorithm, both of which are slightly better than those from the greedy algorithm with no visibility updating or prior information. As with the full volumetric model, accentuating the beliefs was required to help convergence.

Next smoothing prior information was incorporated into the model by including neighbouring compatibility terms similar to those used in the full visibility model, see Section 5.3.2. The resulting reconstruction is shown in Fig. 6.5(b). For comparison, results from the known-visibility single depth-map model are shown in Fig. 6.5(d).

As observed, the resulting reconstruction is not as good as that obtained by the single depth-map model. This is possibly partly due to the prior information that was used. The known-visibility model reconstructs an estimate of the visible surfaces of opaque objects, rather than a solid volume. However, the prior information describes the prior probability distribution for a solid scene model. To improve results, the inclusion of more accurate surface priors should be investigated in the future.

## 6.3 DYNAMIC BELIEF PROPAGATION

To improve the modelling of visibility interactions between voxels using the known visibility model, a novel dynamic belief propagation approach is proposed. This work was first presented at the international conference Image and Vision Computing New Zealand (IVCNZ) 2003 [Forne and Hayes 2003]. This approach is essentially the same as standard belief propagation, except the local compatibility functions are iteratively updated to correspond more closely with the current visibility estimate. A similar approach has recently been used by Larsen et al. [2006] for updating the observation terms in multi-camera stereo algorithm using approximate belief propagation. Although similar to the proposed dynamic approach, the updating of observation or data terms is performed quite differently in their work.

With dynamic belief propagation, the updates of the local conditional probability distributions are performed in a similar manner to that used by the greedy and assignment algorithms presented in Section 4.4. Beginning with an initially transparent estimate, surface voxels are progressively assigned as opaque based on their current belief until a complete scene estimate has been formed. However, unlike the greedy algorithm, the assignment of points is reversible, so that voxels may be restored to a transparent state.

To begin, the local compatibility functions are calculated assuming all surfaces are visible to all cameras. Standard belief propagation is then performed on the belief network for a fixed number of iterations. Next, the most likely set of surface points are assigned opaque. This is done by simply selecting those points whose beliefs are above some threshold value. Scene visibilities are then updated along with the associated

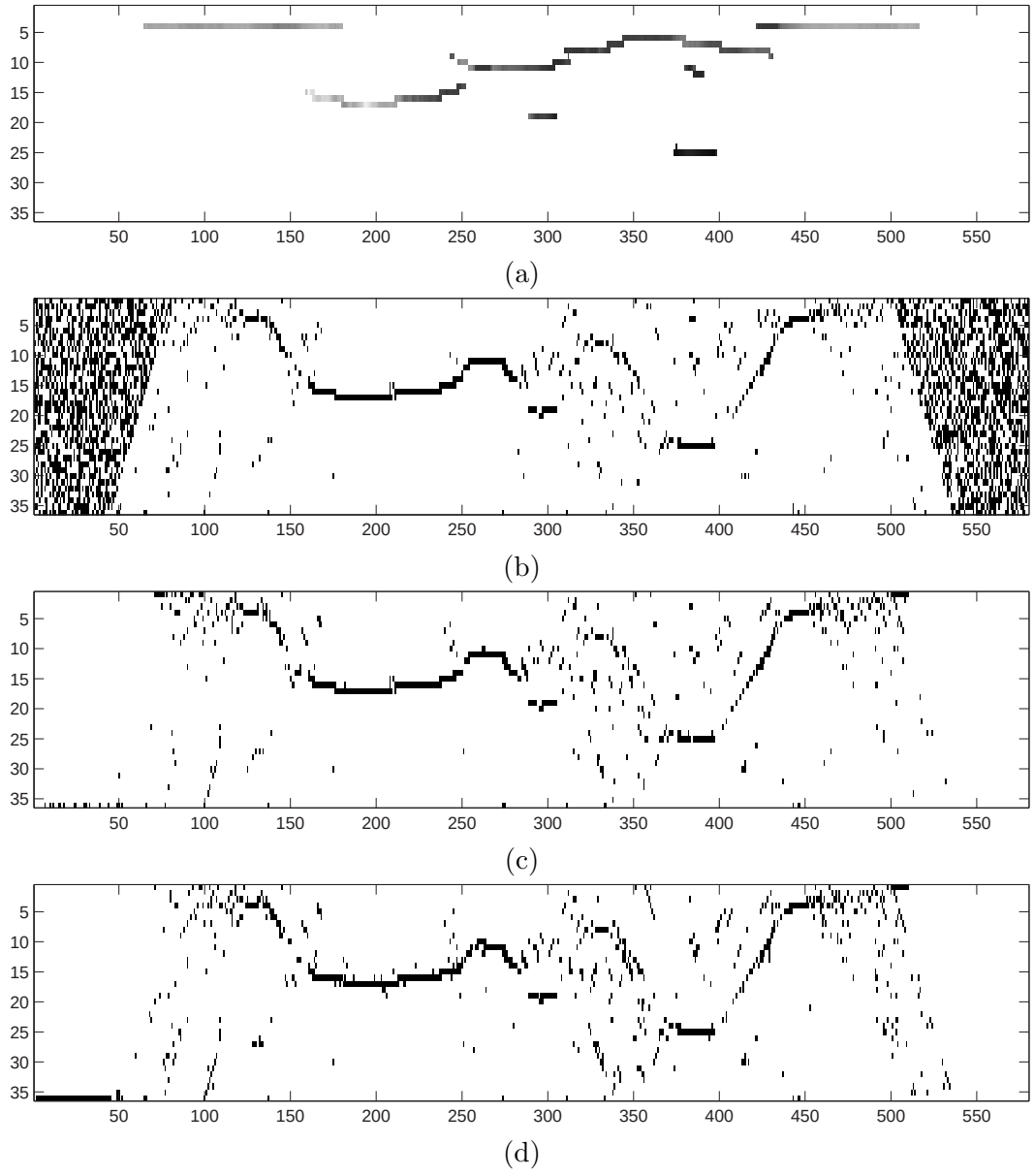


**Figure 6.4** Results on synthetic shapes test set without prior information, showing a horizontal slice through the scene volume. (a) Ideal scene reconstruction. (b) Results from known-visibility model. (c) Results from assignment algorithm. (d) Results from greedy algorithm.

set of local compatibility functions. Following this, belief propagation is continued for another fixed number of iterations. This process is repeated, with the threshold slowly lowered, until a complete estimate of the scene is formed. This has similarities to the process of simulated annealing, where a temperature coefficient is slowly lowered until convergence is achieved.

During each update, the algorithm checks all currently assigned points to see if their belief has fallen below the threshold. If this is the case, the corresponding scene point is restored to a transparent state, thereby allowing decisions to be undone.

### 6.3.1 Results

To evaluate the performance of the dynamic belief propagation approach, and test whether it improves the scene reconstruction over standard fixed visibility belief propagation, the single depth-map and known visibility volumetric models were retested on the synthetic test set using visibility updating. The results from this are shown in Fig. 6.5(c) and (e).

With the known-visibility volumetric model, problems were encountered with selecting the most likely opaque voxels at each iteration. This was a result of having to accentuate the beliefs at each node to ensure convergence. As a consequence, if the beliefs were updated after a fixed number of iterations without accentuating the beliefs the overall system failed to converge properly. If the beliefs were accentuated, they became polarised to either zero or one, and consequently they could not be used to determine the likelihood each voxel was opaque. The resulting reconstruction after 100 iterations without accentuating the beliefs is shown in Fig. 6.5(c). As observed the resulting reconstruction is rather poor.

Initial testing on the single depth-map model highlighted a number of problems, such as the need for visibility priors to ensure the algorithm was stable under visibility updating. These were similar to those used in the greedy algorithm, as described in Section 4.6, and were added to penalise semi-occluded surfaces. The other modification required to prevent instability was to prevent voxels immediately behind assigned opaque voxels from being modelled as occluded. Because of the smoothing and visibility priors, updating the probability of neighbouring voxels immediately behind recently assigned surface voxels resulted in the assigned voxels becoming less likely at the next iteration. This caused the algorithm to become unstable.

With these modifications the results with the single depth-map model were much more promising. As shown in Fig. 6.5(e), the resulting reconstruction appears reasonably similar to the reconstruction obtained without the visibility updating, however, the foreground surfaces are narrower and much closer to those in the ideal model, while more of the background surfaces have been resolved correctly.

To demonstrate the single depth-map dynamic belief propagation algorithm on real

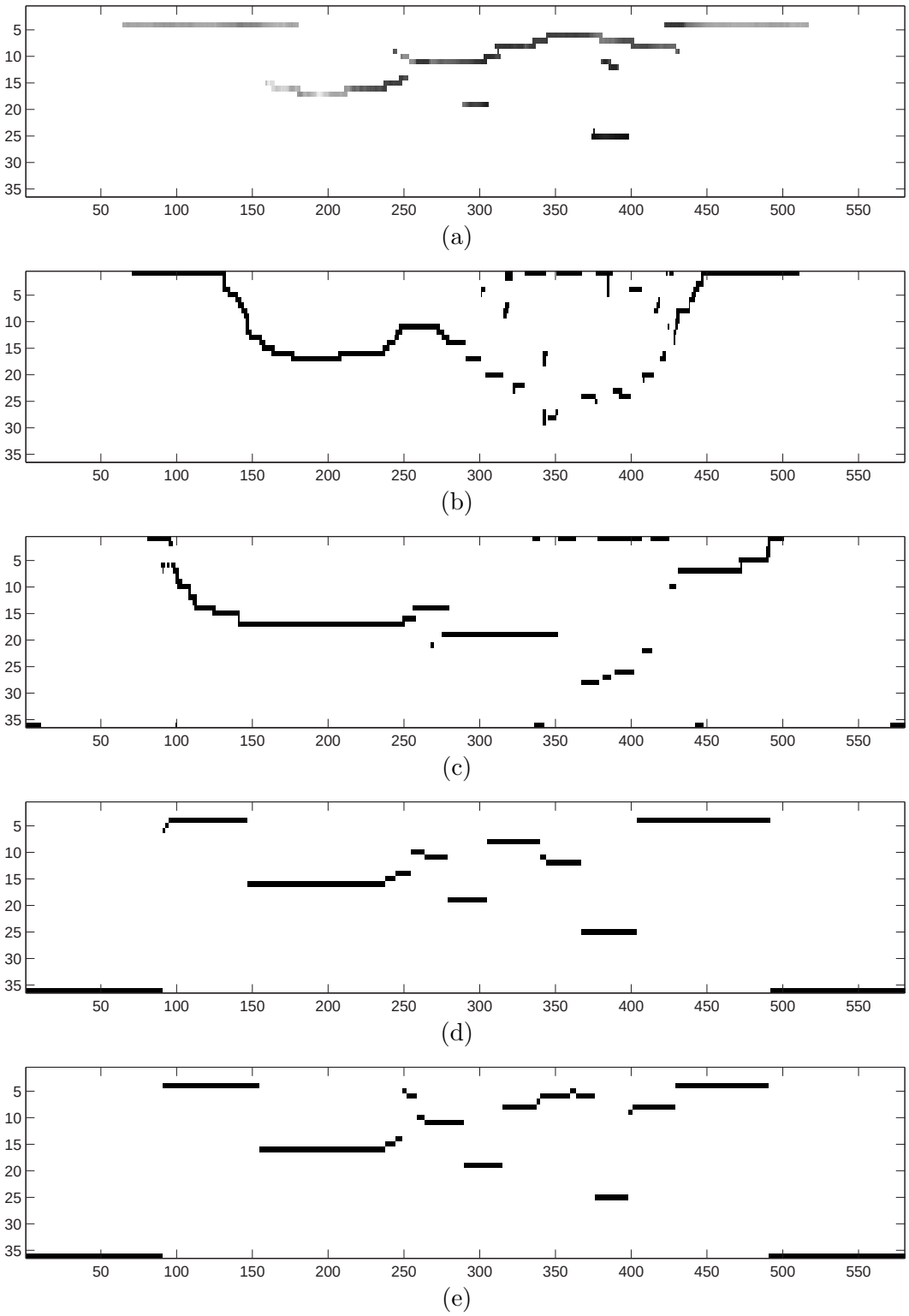


**Figure 6.5** Results of belief propagation with smoothing prior on synthetic shapes test set, showing a horizontal slice through the scene volume at row one hundred. (a) Ideal scene reconstruction. (b) Results from known-visibility volumetric model. (c) Results from dynamic visibility volumetric model. (d) Results from single depth-map model. (e) Results from dynamic visibility single depth-map model.

data, it was applied to the 'Teddy' test set, courtesy of the Middlebury Stereo Vision site[1]. The results are shown in Fig. 6.6(d). For comparison, the results of the single-depth belief propagation algorithm with no visibility updating are shown in Fig. 6.6(c). Although these both appear similar, the resulting depth-map is closer to the ideal depth-map with the dynamic approach. This is highlighted by comparing the errors in the depth-maps, as shown in Fig. 6.6(e) and (f).

One problem with the dynamic approach, is that at each iteration the optimisation is based on the previously obtained visibilities, which may be different from the resulting visibilities. This can lead to local instabilities in some situations, or the scene may converge to a local optimum that is a poor estimate of the global optimum.

## 6.4 DISSIMILARITY MEASURE

As mentioned in Section 2.3, variations in the convolution kernel between cameras can lead to variations in the observed pixel intensities. These intensity variations, if not accounted, cause the conditional probability distribution of obtaining the image data given a particular scene voxel to be miss-calculated. To help alleviate these errors, robust matching measures such as the pixel dissimilarity measure presented by Birchfield and Tomasi [1998b], or mutual information [Hirschmuller 2005, Kim et al. 2003], can be used.

With a discrete scene model the effect of these variations can be reduced by ensuring that the scene sample points are on or near the object surface and that the cameras observe the surface from approximately the same angle. Since the scene is unknown prior to reconstruction, this can be achieved by using a finer sample spacing and only comparing the intensities between cameras with a similar direction of view. The problem with this approach is the memory and computation time of obtaining the scene reconstruction are increased.

An alternative approach is to interpolate between adjacent samples and then take the best match within half the sample spacing either side of a pixel [Birchfield and Tomasi 1998b, Birchfield and Tomasi 1998a]. This is based on the assumption that the images are adequately sampled, so that no aliasing occurs. Although sometimes untrue, this can easily be enforced by introducing additional focal blur. Sampling variations can also be treated as additional system noise correlated with the intensity differences between adjacent pixels.

One of the problems with the pixel dissimilarity measure presented by Birchfield and Tomasi [1998b] is that it does not extend easily to multiple cameras without using a reference image. An alternative symmetric dissimilarity measure is suggested by Szeliski and Scharstein [2002], however, this is also only defined for a pair of camera images. To

[1]See http://vision.middlebury.edu/stereo/

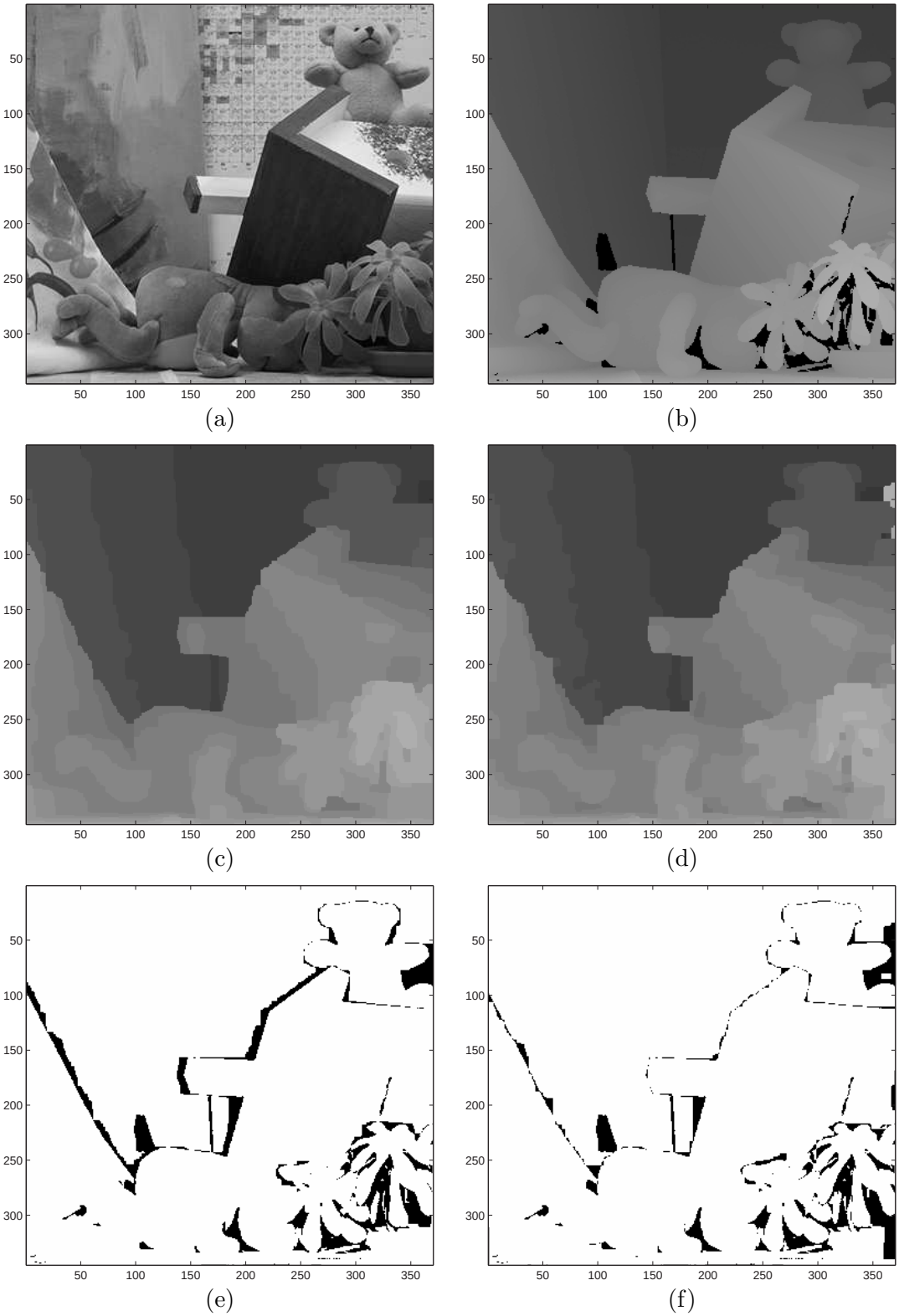


**Figure 6.6** Results from dynamic belief propagation algorithm on the teddy test set. (a) Central image from sequence of five images used. (b) Ideal depth-map. (c) Results from single depth-map belief propagation. (d) Results from single depth-map dynamic belief propagation. (e) Disparity error between reconstructed and ideal depth-maps for single depth-map belief propagation. Dark regions show a disparity error greater or equal to one. (f) Disparity error between reconstructed and ideal depth-maps for single depth-map dynamic belief propagation.

try and improve the reliability of matching pixels in a multiple camera system a novel pixel dissimilarity measure is presented in Section 6.4.1.

### 6.4.1 Multiple camera dissimilarity measure

Given a voxel scene model with binary opacities the depth of a surface can only be resolved up to the sample spacing between voxels. However, variation in surface depth within the voxel spacing will lead to changes in the observed image intensities. These changes can be expressed in terms of the depth of the surface, with the dissimilarity measure being minimised when the estimated surface depth corresponds with the true surface depth.

Consequently, the dissimilarity measure between observed intensities corresponding to a given scene voxel is obtained by searching for a minimum over the range of depths between adjacent samples. This is demonstrated in Fig. 6.7, where the effect of changing the depth of the sample point corresponds with a related change in the projected image position. Assuming linear interpolation, the minimum point can be found by minimising a quadratic term. Using this approach a multiple camera pixel dissimilarity measure can be expressed as

$$\psi_i = \min_{z_i^- < z < z_i^+} \sum_{j \in \Omega_i} \left( I_j(z) - \overline{I}(z) \right)^2 . \tag{6.6}$$

Assuming linear interpolation, the intensity function in each camera can be expressed as

$$I_j(z) = (I_j^+ - I_j^-)\acute{z} + I_j^- , \tag{6.7}$$

where $\acute{z}$ is the normalised change in depth, given by

$$\acute{z} = \frac{z - z_i^-}{z_i^+ - z_i^-}. \tag{6.8}$$

Using this expression, the mean intensity function is given by

$$\begin{aligned} \overline{I}(z) &= \frac{\sum_{j \in \Omega_i} I_j(z)}{|\Omega_i|}, \\ &= \frac{\sum_{j \in \Omega_i} \left( I_j^+ - I_j^- \right)}{|\Omega_i|} \acute{z} + \frac{\sum_{j \in \Omega_i} I_j^-}{|\Omega_i|}, \\ &= \left( \overline{I}^+ - \overline{I}^- \right) \acute{z} + \overline{I}^- , \end{aligned} \tag{6.9}$$

where $\Omega_i$ is the set of images which can observe voxel $i$. Substituting Eq. 6.7 and Eq. 6.9

into Eq. 6.6, the pixel dissimilarity measure can be expressed as

$$\begin{aligned}\psi_i &= \min_{0<\acute{z}<(z_i^+ - z_i^-)} \sum_{j\in\Omega_i} \left( \left( \acute{I}_j^+ - \acute{I}_j^- \right) \acute{z} + \acute{I}_j^- \right)^2, \\ &= \min_{0<\acute{z}<(z_i^+ - z_i^-)} \left( \sum_{j\in\Omega_i} \left( \acute{I}_j^+ - \acute{I}_j^- \right)^2 \acute{z}^2 + 2 \sum_{j\in\Omega_i} \left( \acute{I}_j^+ - \acute{I}_j^- \right) \acute{I}_j^- \acute{z} + \sum_{j\in\Omega_i} \left( \acute{I}_j^- \right)^2 \right), \quad (6.10)\end{aligned}$$

where $\acute{I}_j^+ = (I_j^+ - \overline{I}^+)$ and $\acute{I}_j^- = (I_j^- - \overline{I}^-)$. This function is a quadratic in $\acute{z}$. By differentiating, the minimum depth is given by

$$\acute{z} = \frac{-\sum_{j\in\Omega_i} \left( \acute{I}_j^+ - \acute{I}_j^- \right) \acute{I}_j^-}{\sum_{j\in\Omega_i} \left( \acute{I}_j^+ - \acute{I}_j^- \right)^2}. \quad (6.11)$$

If this is in the range zero to one, then the minimum dissimilarity is found by substituting this expression back into Eq. 6.10, to give

$$\psi_i = \sum_{j\in\Omega_i} \left( \acute{I}_j^- \right)^2 - \frac{\left( \sum_{j\in\Omega_i} \left( \acute{I}_j^+ - \acute{I}_j^- \right) \acute{I}_j^- \right)^2}{\sum_{j\in\Omega_i} \left( \acute{I}_j^+ - \acute{I}_j^- \right)^2}, 0 < \acute{z} < 1. \quad (6.12)$$

If Eq. 6.11 is less than zero or greater than one, then the constrained minimum will occur at one of the boundaries, and is given by

$$\psi_i = \begin{cases} \sum_{j\in\Omega_i} \left( \acute{I}_j^- \right)^2 & \acute{z} < 0 \\ \sum_{j\in\Omega_i} \left( \acute{I}_j^+ \right)^2 & \acute{z} > 1 \end{cases}. \quad (6.13)$$

### 6.4.2 Results

To demonstrate the performance of the multiple camera dissimilarity measure a synthetic test scene was generated consisting of a single plane, with an intensity distribution as shown in Fig. 6.8(a). Five images of this test scene were created from different viewpoints. Using these images the pixel dissimilarities were calculated over a horizontal slice through the scene volume.

The results for the standard sum of square dissimilarity are shown in Fig. 6.8(c). The calculated dissimilarity gives a poor measure of the correspondence even for a number of correct voxels along the true surface. In comparison, the proposed multiple camera pixel dissimilarity measure gives a good correspondence for all voxels along the true surface as shown in Fig. 6.8(e). The results from the dissimilarity measure proposed by Birchfield and Tomasi [1998b] are also compared in Fig. 6.8(g). As shown, this dissimilarity also gives good correspondences along the true surface.

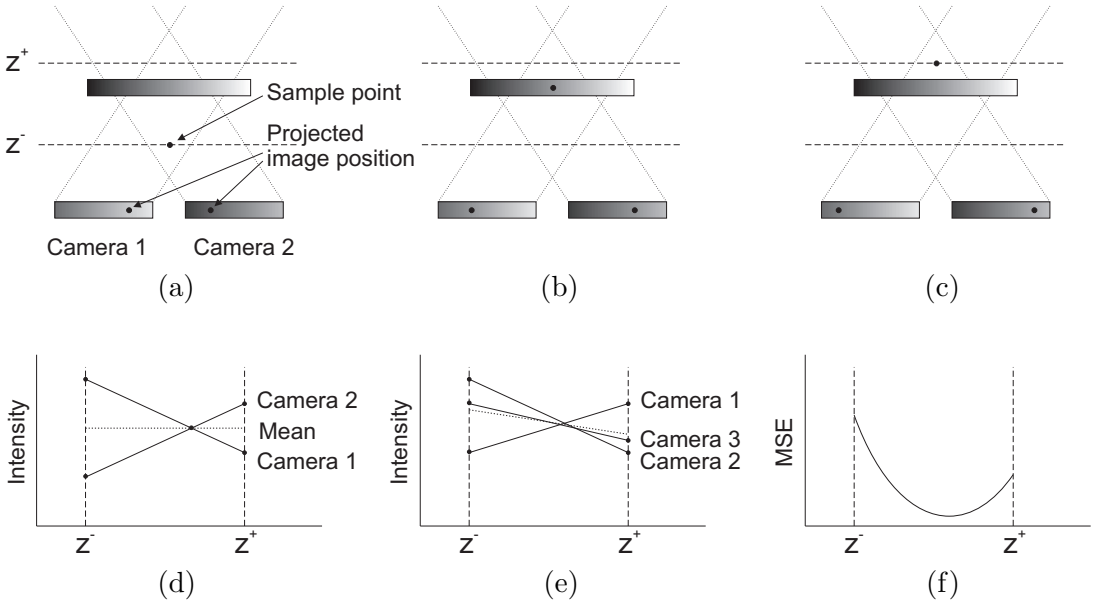


**Figure 6.7** The displacement of sample points from the true surface affects the observed similarity between the pixel intensities which correspond with that point. As shown in (a)–(c), the projected image position and observed intensity of a scene voxel vary with depth, with the minimum dissimilarity between observed intensities occurring the true surface depth. This intensity variation can be plotted as a function of depth (d). (e) With three or more cameras the image intensities are unlikely to agree exactly at any depth, but should correspond closely near the true surface depth. The resulting MSE as a function of depth for three cameras is given in (f). Assuming linear interpolation, the minimum dissimilarity can be found analytically, using Eq. 6.12 and Eq. 6.13.

To demonstrate some of the limitations with the assumption of linear interpolation, the three pixel dissimilarity measures were retested on the same test scene, except with high frequency intensity variations along the surface. The results are shown in Fig. 6.8(b,d,f,h). As shown the linearity assumption made by the multiple camera pixel dissimilarity measure becomes less accurate at higher frequencies, resulting in a number of poorly estimated correspondences. The approach by Birchfield and Tomasi [1998b] was demonstrated to be relatively insensitive to this problem.

To show one of the advantages of the multiple camera pixel dissimilarity measure over the two camera approach [Birchfield and Tomasi 1998b], the dissimilarity measure along the true surface was calculated with the addition of white and coloured noise with a 20 dB signal to noise ratio. The results are plotted in Fig. 6.9. This shows an improved calculation of the true correspondence of this approach over that presented by Birchfield and Tomasi [1998b] for low frequency noise, while the results with higher frequency noise were similar with both approaches. The improved performance with low frequency noise is related to the equal treatment of all images, preventing errors in a single image from significantly affecting the solution.

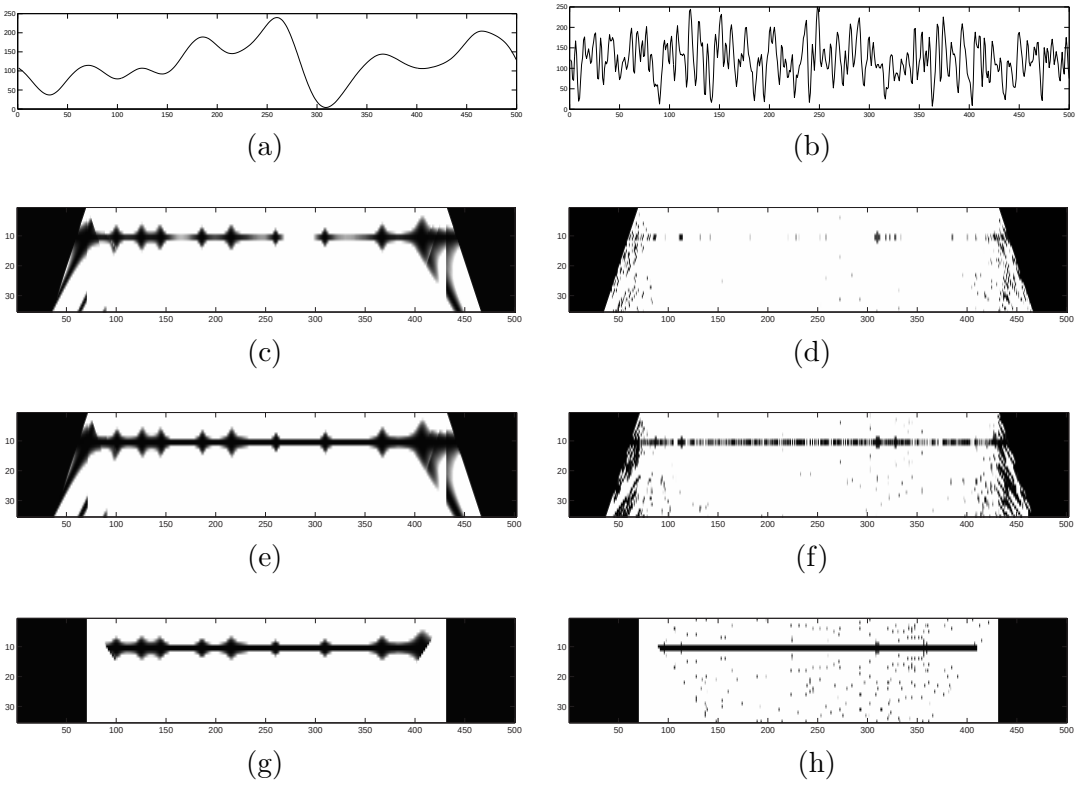


**Figure 6.8** Results of pixel dissimilarity measure on synthetic test scene. (a) and (b) show plots of the surface radiances modelling low and high frequency variations. (c) and (d) show the resulting pixel dissimilarity measure obtained using standard sum of projected square errors. (e) and (f) show the resulting pixel dissimilarity measure for the proposed multiple camera pixel dissimilarity measure. (g) and (h) show the resulting pixel dissimilarity measure for pair-wise pixel dissimilarity measure proposed by Birchfield and Tomasi [1998b].

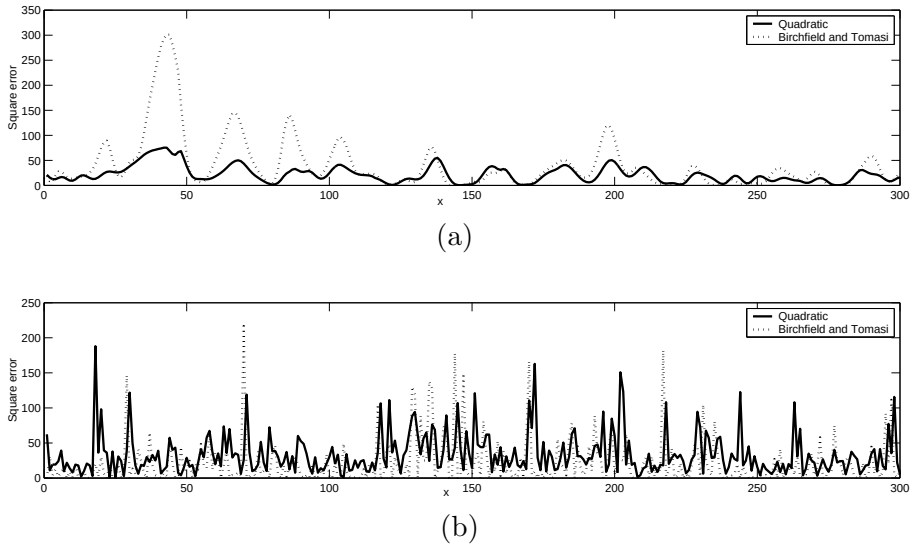


**Figure 6.9** The sensitivity of the proposed multiple camera pixel dissimilarity measure (Quadratic) to system noise was compared to that of the pair-wise dissimilarity measure proposed by Birchfield and Tomasi [1998b]. This was demonstrated on (a) low frequency correlated noise and (b) white noise.

# Chapter 7

## CONCLUSION

The image formation process was described in Chapter 2. Here an original model for the scene reconstruction problem was presented, describing the relationship between camera data and a discrete model of the scene. This highlighted some of the problems with the binary opacity assumption, showing this can cause problems around oblique surfaces or at depth discontinuities. It was also shown that the formation of pixel intensities could be approximated by an integral or discrete summation of the band limited scene opacities and radiances along a line in space, simplifying the more exact 3D integral.

Additionally, the concept of the imaging convolution kernel was considered. It was shown that differences in the convolution kernels lead to variations in observed intensities between images. An alternative multiple camera approach for dealing with these variations was later presented in Chapter 6.

An overview of reconstruction techniques was provided in Chapter 3. The use of a statistical approach to deal with the scene reconstruction problem was introduced and the differences between MAP and MMSE were highlighted. It was discussed how traditional stereo matching could be performed using a volumetric scene model.

Chapter 3 also demonstrated that region based matching and many of the techniques for dealing with non-Lambertian surfaces were equivalent to filtering individual voxel likelihoods within a volumetric model. Techniques for dealing with visibility interactions were introduced, and a variety of optimisation techniques which could be applied to the scene reconstruction problem were discussed.

The problems posed by occlusions were dealt with in Chapter 4. It was demonstrated that the joint probability distribution could be expressed as a product of independent terms, if the visibility of the opaque voxels was known. This allowed the MAP estimate to be expressed as a summation of independent data error terms corresponding with the negative conditional log probabilities. The concept of a complete scene estimate was introduced. This was defined as an estimate where at least one opaque voxel or surface is required along each pixel ray in every image. The summation of independent data error terms could then be formulated as a pixel ray assignment problem, where the objective was to assign at least one opaque voxel along every pixel ray so that

the sum of error terms was a minimum.

In Chapter 4 an iterative approach for dealing with visibility updating was also introduced. Here it was shown that a greedy selection process was more appropriate with visibility updating than the described pixel ray assignment algorithm.

In the work of Preddey and Lane [1997] and Harding et al. [2000] the assignment of voxels was based solely on the projected square error of each voxel. In this work, some simple and improved techniques for reliably assigning opaque voxels were developed. Prior information was used to assist opaque voxel assignment. The results in this chapter highlighted some of the problems with the binary opacity model, motivating the development of a pixel dissimilarity measure presented in Chapter 6. It was also found that a combination of both smoothing and visibility priors produced the best results. A hierarchical approach for efficiently calculating the most likely voxel at each iteration was presented.

To improve the use of prior information and obtain a global optimum, belief propagation was applied. An improved volumetric model of the scene was presented to model the visibility interaction between scene variables and incorporate prior information. This was represented using a factor graph model describing the joint probability of the imaging system. It was shown that belief propagation could be applied to this model to find the MAP estimate of the scene. The local structure of the probability distribution within the model was utilised to compute the message updating more efficiently for this particular volumetric model. However, the resulting algorithm was found to be unstable and a simple technique for helping convergence was developed.

The results in Chapter 5 were promising, but the model was very memory intensive and computing messages at each iteration time consuming, in part due to the volumetric nature of the model. Nevertheless, one reason for investigating belief propagation over other optimisation techniques is that it is highly parallelisable, lending itself to efficient implementation on parallel architectures.

To avoid some of the problems encountered with the volumetric model presented in Chapter 5, a dynamic approach to modelling the scene was presented, where the local probability distributions were iteratively updated to reflect the visibility between scene variables. To ensure the scene estimate was complete a new known-visibility volumetric model was presented. However, it was found to be unstable using the max product belief propagation algorithm to optimise the model. The dynamic updating was also applied to an alternative simpler single depth map model, with promising results, showing that this approach can be used to improve the scene reconstruction process.

## 7.1 RECOMMENDATIONS FOR FUTURE RESEARCH

Future work will focus on improving the system model, as well as attempting to increase the speed of the existing algorithm.

### 7.1.1 Improving system modelling

Work could be done to improve modelling of the imaging system, or techniques for identifying the errors that occur during modelling could be developed. The most significant problems are sampling variations caused by differences in the imaging convolution kernel, as well as problems with the binary opacity assumption in the vicinity of steeply sloping surfaces or depth discontinuities. Specular reflections are another common cause of reconstruction error.

Variations between the modelled and observed intensity of points degrade the reconstruction if not properly accounted. Opportunity for future research is significant. Some ideas include identifying depth discontinuities and adjusting the modelled joint probability distribution accordingly. The modelled noise level could then be reduced as the system model is improved. An improved pixel dissimilarity measure for multiple cameras also needs to be developed, possibly taking into account the local slope in the observed intensity functions.

### 7.1.2 Developing application of prior information

Improved results can also be obtained through more detailed application of prior information. This has been observed in the performance of recent reconstruction algorithms. For example, on the Middlebury test set[1] all the currently top-performing algorithms use some form of image segmentation based on the relationship between observed intensity and scene structure. As applied to the volumetric models presented in this thesis, surface priors may prove more effective than volumetric priors.

### 7.1.3 Improving global optimisation techniques

Another place for significant improvement is in the use of more effective and efficient global optimisation techniques for optimising the joint probability distribution. Recent techniques such as tree re-weighted message passing and graph cuts could prove more successful depending on the structure of the joint probability distribution. Continuous optimisation techniques such as expectation propagation could also be considered. Perhaps the best approach will be to use one technique to form a rough global optimum and then refine this using a stronger but more local optimisation technique.

### 7.1.4 Efficient implementation

Improvements could also be made to the efficiency of optimisation algorithms through better implementation and custom hardware. Implementation of belief propagation on custom hardware would allow the computation of messages to be performed in parallel.

[1] See http://vision.middlebury.edu/stereo/

It should be noted that the belief propagation can be performed in the logarithmic domain thereby replacing multiplication with addition. This, in conjunction with parallel implementation, makes it suitable for fast hardware implementation.

For sequential implementation on a standard PC, the efficiency of the belief propagation algorithm could be improved by appropriate synchronisation of the message updating, or performing more updates in regions where beliefs are changing rapidly.

# Appendix A

## CONVOLUTION EQUIVALENCE

To prove that the pixel intensities given in Theorem 1 will remain the same after convolving the integrand with a normalised depth invariant window function in the $Z$ direction, the resulting integral is shown to be equivalent, assuming $D_i(u,v,Z)T_i(u,v,Z) = 0$ close to the camera.

Beginning with the expression for pixel intensities given in Theorem 1,

$$C_i(x,y) = \int_{Z=0}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D_i(u,v,Z)T_i(u,v,Z)h_i(x,y,u,v)\,\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}Z + N_i(x,y), \quad \text{(A.1)}$$

and using vector variable $t = (u,v,w)$, the integrand can be convolved with the normalised depth invariant window function $\gamma_i(x,y,u,v,Z-w)$ in the $Z$ direction, to give

$$C_i(x,y) = \int_{Z=0}^{\infty}\iiint_{-\infty}^{\infty} D_i(t)T_i(t)h_i(x,y,u,v)\gamma_i(x,y,u,v,Z-w)\,\mathrm{d}t\,\mathrm{d}Z \quad \text{(A.2)}$$

Using $u(Z)$ to represent the unit step function in the $Z$ direction, this can be rearranged to give,

$$\begin{aligned} C_i(x,y) &= \iiint_{-\infty}^{\infty} D_i(t)T_i(t)h_i(x,y,u,v)\int_{Z=0}^{\infty}\gamma_i(x,y,u,v,Z-w)\,\mathrm{d}Z\,\mathrm{d}t \\ &= \iiint_{-\infty}^{\infty} D_i(t)T_i(t)h_i(x,y,u,v)\int_{-\infty}^{\infty}\gamma_i(x,y,u,v,Z-w)u(Z)\,\mathrm{d}Z\,\mathrm{d}t. \end{aligned} \quad \text{(A.3)}$$

Since $\gamma_i(x,y,u,v,Z-w)$ is normalised so that the infinite integral over $\gamma_i(x,y,u,v,Z-w)$ in the $Z$ direction equals one, the inner integral in Eq. A.3 can be expressed as,

$$\int_{-\infty}^{\infty}\gamma_i(x,y,u,v,Z-w)u(Z)\,\mathrm{d}Z = \int_{-\infty}^{\infty}\gamma_i(x,y,u,v,Z)u(Z+w)\,\mathrm{d}Z \quad \text{(A.4)}$$

$$= \begin{cases} 0 & w < -\gamma_{i\max}(x,y,u,v) \\ 1 & w > -\gamma_{i\min}(x,y,u,v) \\ \int_{-w}^{\infty}\gamma_i(x,y,u,v,Z)\,\mathrm{d}Z & \text{otherwise,} \end{cases} \quad \text{(A.5)}$$

where $\gamma_{i\min}(x,y,u,v)$ and $\gamma_{i\max}(x,y,u,v)$ are the minimum and maximum extents of

$\gamma_i(x, y, u, v, Z)$ in the $Z$ direction. Substituting Eq. A.5 into Eq. A.3, the average pixel intensities can be rewritten as,

$$C_i(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{w=-\gamma_{i\min}(x,y,u,v)}^{-\gamma_{i\max}(x,y,u,v)} D_i(t)T_i(t)h_i(x,y,u,v)\int_{-w}^{\infty}\gamma_i(x,y,u,v,Z)\,\mathrm{d}Z\,\mathrm{d}w\,\mathrm{d}u\,\mathrm{d}v + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{w=-\gamma_{i\min}(x,y,u,v)}^{\infty} D_i(t)T_i(t)h_i(x,y,u,v)\,\mathrm{d}w\,\mathrm{d}u\,\mathrm{d}v. \quad \text{(A.6)}$$

Now, assuming $D_i(u, v, w)T_i(u, v, w) = 0$ if $-\gamma_{i\max}(x, y, u, v) < w < -\gamma_{i\min}(x, y, u, v)$, the second term in Eq. A.6 will also equal zero. Therefore, the pixel intensities can be simplified to

$$C_i(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{w=-\gamma_{i\min}(x,y,u,v)}^{\infty} D_i(t)T_i(t)h_i(x,y,u,v)\,\mathrm{d}w\,\mathrm{d}u\,\mathrm{d}v. \quad \text{(A.7)}$$

Using the property $D_i(u, v, w)T_i(u, v, w) = 0$ if $w < -\gamma_{i\min}(x, y, u, v)$, the limits of integration can be changed from $w = -\gamma_{i\min}(x, y, u, v)$ to $w = 0$. With this substitution, and a change of coordinates from $w$ to $Z$, the pixel intensities can finally be expressed as

$$C_i(x,y) = \int_{Z=0}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} D_i(u,v,Z)T_i(u,v,Z)h_i(x,y,u,v)\,\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}Z, \quad \text{(A.8)}$$

which is equal to the original expression for pixel intensities.

# Appendix B

## PROBABILITY UPDATING

Consider the simple stereo model shown in Fig. B.1. This models the joint probability distribution of a single scene point $S_1$ with radiance $R_1$ and opacity $O_1$, and the corresponding intensities $I = \{I_1, I_2, \ldots, I_N\}$ observed along pixel rays passing through this point. Assuming that scene points are either completely opaque or transparent, the opacity and visibility can be modelled as binary variables. Also, by assuming that the prior probability of opacity is constant and the prior probability of radiance is uniform over the range $R_{min}$ to $R_{max}$, these priors can be written as,

$$P(O_1) = \begin{cases} \frac{1}{2} & \text{if } O_1 = \text{opaque} \\ \frac{1}{2} & \text{if } O_1 = \text{transparent} \end{cases} \tag{B.1}$$

$$P(R_1) = \begin{cases} \frac{1}{K} & \text{if } R_{\min} < R_1 < R_{\max} \\ 0 & \text{otherwise} \end{cases}, \tag{B.2}$$

where $K = R_{\max} - R_{\min}$. Modelling the image noise distribution as a Gaussian function with variance $\sigma^2$, the conditional probability of pixel $I_i$ can be written as,

$$P(I_i|O_1, R_1) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(I_i - R_1)^2}{2\sigma^2} & \text{if } O_1 = \text{opaque} \\ \frac{1}{K} & \text{if } O_1 = \text{transparent} \end{cases} \tag{B.3}$$

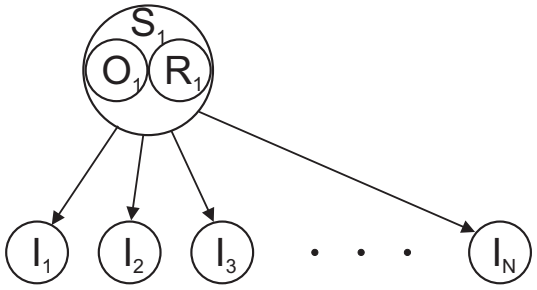


**Figure B.1** Probabilistic model of imaging system, describing the statistical relationship between the observed image intensities $I = \{I_1, I_2, \ldots, I_N\}$ and the opacity $O_1$ and radiance $R_1$ of a scene point $S_1$.

Using the MAP approach, the scene reconstruction problem can be expressed as determining the opacity of $S_1$ which is most likely given images $I$. From Bayes' rule the MAP estimate can be expressed as,

$$\arg\max_{O_1} P(O_1|I) = \arg\max_{O_1} \frac{P(I|O_1)P(O_1)}{P(I)} \tag{B.4}$$

Here $P(I|O_1)$ is the conditional probability of obtaining the images given the opacity of $S_1$, $P(O_1)$ is the prior probability that $S_1$ is opaque, and $P(I)$ is the prior probability of obtaining the images $I$. By applying the "summation" rule, the term $P(I|O_1)$ can be expanded to give,

$$\begin{aligned} P(I|O_1) &= \int_{R_1=-\frac{-K}{2}}^{\frac{K}{2}} P(I, R_1|O_1)\, dR_1 \\ &= \int_{R_1=-\frac{-K}{2}}^{\frac{K}{2}} P(I|R_1, O_1)P(R_1|O_1)\, dR_1 \\ &= \int_{R_1=-\frac{-K}{2}}^{\frac{K}{2}} P(I|S_1)P(R_1|O_1)\, dR_1 \end{aligned} \tag{B.5}$$

Further, if the image noise in each sensor is independent, the first term within the integral can be rewritten as,

$$\begin{aligned} P(I|S_1) &= \prod_{i=1}^{N} P(I_i|S_1) \\ &= \begin{cases} (\frac{1}{\sigma\sqrt{2\pi}})^N \prod_{i=1}^{N} \exp\frac{-(I_i-R_1)^2}{2\sigma^2} & \text{if } O_1 = \text{opaque} \\ (\frac{1}{K})^N & \text{otherwise} \end{cases} \end{aligned} \tag{B.6}$$

The second term, $P(R_1|O_1)$, can also be simplified since $R_1$ and $O_1$ are independent, giving

$$\begin{aligned} P(R_1|O_1) &= P(R_1) \\ &= \begin{cases} \frac{1}{K} & \text{if } \frac{-K}{2} < R_1 < \frac{K}{2} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{B.7}$$

Substituting Eq. B.6 and Eq. B.7 back into Eq. B.5 and moving constants outside the integral gives,

$$P(I|O_1) = \begin{cases} \frac{1}{K}(\frac{1}{\sigma\sqrt{2\pi}})^N \int_{R_1=\frac{-K}{2}}^{\frac{K}{2}} \prod_{i=1}^{N} \exp\frac{-(I_i - R_1)^2}{2\sigma^2}\, dR_1 & \text{if } O_1 = \text{opaque} \\ (\frac{1}{K})^N & \text{otherwise} \end{cases} \tag{B.8}$$

For the case where $O_1$ is opaque, the integral can be evaluated by first rewriting the

integrand in the form of a one-dimensional Gaussian type function. Using the Product Rule for exponents the integrand can be expressed as,

$$\prod_{i=1}^{N} \exp \frac{-(I_i - R_1)^2}{2\sigma^2} = \exp - \sum_{i=1}^{N} \frac{(I_i - R_1)^2}{2\sigma^2} \tag{B.9}$$

The exponent is then expanded out to give,

$$\begin{aligned} -\sum_{i=1}^{N} \frac{(I_i - R_1)^2}{2\sigma^2} &= -\left( \sum_{i=1}^{N} \frac{I_i^2}{2\sigma^2} - \sum_{i=1}^{N} \frac{2R_1 I_i}{2\sigma^2} + \sum_{i=1}^{N} \frac{R_1^2}{2\sigma^2} \right) \\ &= \frac{-1}{2\sigma^2} \left( \sum_{i=1}^{N} I_i^2 - 2R_1 \sum_{i=1}^{N} I_i + N R_1^2 \right) \\ &= \frac{-1}{2\sigma^2} \left( \sum_{i=1}^{N} I_i^2 - 2R_1 N \overline{I} + N R_1^2 \right) \\ &= \frac{-1}{2\sigma^2} \left( \sum_{i=1}^{N} I_i^2 + N(R_1 - \overline{I})^2 - N\overline{I}^2 \right) \\ &= \frac{-N(R_1 - \overline{I})^2}{2\sigma^2} - \frac{1}{2\sigma^2} \left( \sum_{i=1}^{N} I_i^2 - N\overline{I}^2 \right), \end{aligned} \tag{B.10}$$

where $\overline{I}$ is the mean value of I. Substituting Eq. B.9 and Eq. B.10 back into Eq. B.8, and rearranging for the case where $O_1$ = opaque gives,

$$\begin{aligned} P(I|O_1) &= \frac{1}{K} (\frac{1}{\sigma\sqrt{2\pi}})^N \int_{R_1=\frac{-K}{2}}^{\frac{K}{2}} \exp \left( \frac{-N(R_1 - \overline{I})^2}{2\sigma^2} - \frac{1}{2\sigma^2} \left[ \sum_{i=1}^{N} I_i^2 - N\overline{I}^2 \right] \right) dR_1 \\ &= \frac{1}{K} (\frac{1}{\sigma\sqrt{2\pi}})^N \int_{R_1=\frac{-K}{2}}^{\frac{K}{2}} \exp \frac{-(R_1 - \overline{I})^2}{\frac{2\sigma^2}{N}} \exp \left( -\frac{1}{2\sigma^2} \left[ \sum_{i=1}^{N} I_i^2 - N\overline{I}^2 \right] \right) dR_1 \\ &= \frac{1}{K} (\frac{1}{\sigma\sqrt{2\pi}})^N \exp \left( -\frac{1}{2\sigma^2} \left[ \sum_{i=1}^{N} I_i^2 - N\overline{I}^2 \right] \right) \int_{R_1=\frac{-K}{2}}^{\frac{K}{2}} \exp \frac{-(R_1 - \overline{I})^2}{\frac{2\sigma^2}{N}} dR_1 \end{aligned} \tag{B.11}$$

The integrand is now a Gaussian function which can be easily evaluated to give,

$$\int_{R_1=\frac{-K}{2}}^{\frac{K}{2}} \exp \frac{-(R_1 - \overline{I})^2}{\frac{2\sigma^2}{N}} dR_1 \approx \int_{R_1=-\infty}^{\infty} \exp \frac{-(R_1 - \overline{I})^2}{\frac{2\sigma^2}{N}} dR_1 = \frac{\sigma\sqrt{2\pi}}{\sqrt{N}}, \tag{B.12}$$

assuming $\overline{I}$ sufficiently distant from $\frac{-K}{2}$ or $\frac{K}{2}$ so as to avoid truncation effects. Substituting this back into Eq. B.11 gives,

$$\begin{aligned} P(I|O_1) &= \frac{1}{K} (\frac{1}{\sigma\sqrt{2\pi}})^N \exp \left( -\frac{1}{2\sigma^2} \left[ \sum_{i=1}^{N} I_i^2 - N\overline{I}^2 \right] \right) \frac{\sigma\sqrt{2\pi}}{\sqrt{N}} \quad \text{if } O_1 = \text{opaque} \\ &= \frac{\sigma\sqrt{2\pi}}{K\sqrt{N}(\sigma\sqrt{2\pi})^N} \exp \left( -\frac{1}{2\sigma^2} \left[ \sum_{i=1}^{N} I_i^2 - N\overline{I}^2 \right] \right) \quad \text{if } O_1 = \text{opaque} \end{aligned} \tag{B.13}$$

Finally the exponent can be rewritten as,

$$\begin{aligned}
\frac{-1}{2\sigma^2}\left(\sum_{i=1}^{N} I_i^2 - N\overline{I}^2\right) &= \frac{-1}{2\sigma^2}\left(\sum_{i=1}^{N} I_i^2 - 2N\overline{I}^2 + N\overline{I}^2\right) \\
&= \frac{-1}{2\sigma^2}\left(\sum_{i=1}^{N} I_i^2 - 2\overline{I}N\overline{I} + N\overline{I}^2\right) \\
&= \frac{-1}{2\sigma^2}\left(\sum_{i=1}^{N} I_i^2 - 2\overline{I}\sum_{i=1}^{N} I_i + \sum_{i=1}^{N}\overline{I}^2\right) \\
&= \frac{-1}{2\sigma^2}\sum_{i=1}^{N}\left(I_i^2 - 2\overline{I}I_i + \overline{I}^2\right) \\
&= \frac{-1}{2\sigma^2}\sum_{i=1}^{N}(I_i - \overline{I})^2 \\
&= \sum_{i=1}^{N}\frac{-(I_i - \overline{I})^2}{2\sigma^2} \qquad \text{(B.14)}
\end{aligned}$$

giving,

$$P(I|O_1) = \begin{cases} \frac{\sigma\sqrt{2\pi}}{K\sqrt{N}}\prod_{i=1}^{N}\frac{1}{\sigma\sqrt{2\pi}}\exp\frac{-(I_i-\overline{I})^2}{2\sigma^2} & \text{if } O_1 = \text{opaque} \\ (\frac{1}{K})^N & \text{otherwise} \end{cases} \qquad \text{(B.15)}$$

The next step, in calculating the posteriori probability that $S_1$ is opaque given observations I, is to calculate the prior probability of obtaining the images $P(I)$. Using Eq. B.15 this can be expressed as,

$$\begin{aligned}
P(I) &= \sum_{O_1} P(I, O_1) \\
&= \sum_{O_1} P(I|O_1)P(O_1) \\
&= \sum_{O_1} P(I|O_1)\frac{1}{2} \\
&= \frac{1}{2}P(I|O_1 = \text{opaque}) + \frac{1}{2}P(I|O_1 = \text{transparent}) \\
&= \frac{1}{2}\frac{\sigma\sqrt{2\pi}}{K\sqrt{N}}\prod_{i=1}^{N}\frac{1}{\sigma\sqrt{2\pi}}\exp\frac{-(I_i-\overline{I})^2}{2\sigma^2} + \frac{1}{2}(\frac{1}{K})^N
\end{aligned} \qquad \text{(B.16)}$$

Substituting Eq. B.15 and Eq. B.16 back into Eq. B.4, the probability that $S_1$ is opaque

or transparent, given images $I$, can be expressed as,

$$P(O_1|I) = \begin{cases} \dfrac{\frac{\sigma\sqrt{2\pi}}{K\sqrt{N}} \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(I_i-\overline{I})^2}{2\sigma^2}}{\frac{\sigma\sqrt{2\pi}}{K\sqrt{N}} \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(I_i-\overline{I})^2}{2\sigma^2} + (\frac{1}{K})^N} & \text{if } O_1 = \text{opaque} \\ \dfrac{(\frac{1}{K})^N}{\frac{\sigma\sqrt{2\pi}}{K\sqrt{N}} \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(I_i-\overline{I})^2}{2\sigma^2} + (\frac{1}{K})^N} & \text{otherwise} \end{cases} \tag{B.17}$$

Having determined the posteriori probability of a scene point's opacity, the MAP estimate can simply be found by selecting the opacity that has the highest probability,

$$\text{MAP} = \arg\max_{O_1} \left( \begin{array}{ll} \frac{\sigma\sqrt{2\pi}}{K\sqrt{N}} \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(I_i-\overline{I})^2}{2\sigma^2} & \text{if } O_1 = \text{opaque} \\ (\frac{1}{K})^N & \text{otherwise} \end{array} \right) \tag{B.18}$$

# REFERENCES


Aji, S.M. and Mceliece, R.J. (2001), 'The generalized distributive law and free energy minimization', In *Proceedings of the 39th Allerton Conference on Communication, Control and Computing*.

Aschwanden, P. and Guggenbuhl, W. (1992), 'Experimental results from a comparative study on correlation-type registration algorithms'.

Baker, S., Szeliski, R. and Anandan, P. (1998), 'A layered approach to stereo reconstruction', In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June, pp. 434–441.

Barclay, P.J., Forne, C.J., Hayes, M.P. and Gough, P.T. (2003), 'Reconstructing seafloor bathymetry with a multichannel broadband inSAS using belief propagation', In *Proceedings of OCEANS 2003*, pp. 2149–2154.

Bellman, R. (1960), 'Sequential machines, ambiguity, and dynamic programming', *Journal of the ACM*, Vol. 7, No. 1, pp. 24–28.

Birchfield, S. and Tomasi, C. (1998a), 'Depth discontinuities by pixel-to-pixel stereo', In *Proceedings of the IEEE International Conference on Computer Vision*, January, pp. 1073–1080.

Birchfield, S. and Tomasi, C. (1998b), 'A pixel dissimilarity measure that is insensitive to image sampling', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 4, pp. 401–406.

Birchfield, S. and Tomasi, C. (1999), 'Multiway cut for stereo and motion with slanted surfaces', In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 489–495.

Bleyer, M. and Gelautz, M. (2007), 'Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions', *Image Communications*, Vol. 22, No. 2, pp. 127–143.

Bones, P.J., Bretschneider, T., Forne, C.J., Millane, R.P. and McNeill, S.J. (2000), 'Tomographic blur identification using image edges', In Fiddy, M.A.

and Millane, R.P. (Eds.), *Proceedings of SPIE, Image Reconstruction from Incomplete Data*, November, pp. 133–141.

Born, M. and Wolf, E. (1980), *Principles of Optics*, Pergamon Press, Oxford, 6th ed.

Boykov, Y., Veksler, O. and Zabih, R. (2001), 'Fast approximate energy minimization via graph cuts', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, November, pp. 1222–1239.

Carr, J.C., Beatson, R.K., McCallum, B.C., Fright, W.R., McLennan, T.J. and Mitchell, T.J. (2003), 'Smooth surface reconstruction from noisy range data', In *Proceedings of ACM Graphite 2003*, February, pp. 119–126.

Chambon, S. and Crouzil, A. (2004), 'Towards correlation-based matching algorithms that are robust near occlusions', In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)*, IEEE Computer Society, Washington, DC, USA, pp. 20–23.

Chen, Q. and Medioni, G. (1999), 'A volumetric stereo matching method: Application to image-based modeling', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Vol. 1.

Cortelazzo, G., Mian, G.A. and Parolari, R. (1994), 'Statistical characteristics of granular camera noise', *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 4, December, pp. 536–543.

Cox, I., Roy, S. and Hingorani, S. (1995), 'Dynamic histogram warping of image pairs for constant image brightness', In *Proceedings of the IEEE International Conference on Image Processing*, pp. 366–369.

Cox, I., Hingorani, S. and Rao, S. (1996), 'A maximum likelihood stereo algorithm', *Computer Vision and Image Understanding*, Vol. 63, No. 3, May, pp. 542–567.

Culbertson, B., Malzbender, T. and Slabaugh, G. (1999), 'Generalized voxel coloring', In *Proceedings of the ICCV'99 Vision Algorithms Workshop*, September.

De Bonet, J.S. and Viola, P.A. (1999), 'Roxels: Responsibility weighted 3D volume reconstruction', In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 418–425.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977), 'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 39, No. 1, pp. 1–38.

Eisert, P., Steinbach, E. and Girod, B. (1999), 'Multi-hypothesis, volumetric reconstruction of 3-D objects from multiple calibrated camera views', In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, March, pp. 3509–3512.

Farid, H., Lee, S.W. and Bajcsy, R. (1994), *View Selection Strategies for Multi-View, Wide-Baseline Stereo*, Technical Report MS-CIS-94-18, Department of Computer and Information Science, University of Pennsylvania.

Faugeras, O.D. and Keriven, R. (1997), 'Level set methods and the stereo problem', In *Scale-Space Theories in Computer Vision*, pp. 272–283.

Faugeras, O.D. and Keriven, R. (1998), 'Variational principles, surface evolution, PDEs, level set methods and the stereo problem', In *IEEE Transactions on Image Processing*, pp. 336–344.

Faugeras, O., Hotz, B., Mathieu, H., Vieville, T., Zhang, Z., Fua, P., Theron, E., Moll, L., Berry, G., Vuillemin, J., Bertin, P. and Proy, C. (1993), *Real time correlation-based stereo: Algorithm, implementations and applications*, Technical Report RR-2013, INRIA.

Felzenszwalb, P.F. and Huttenlocher, D.R. (2004), 'Efficient belief propagation for early vision', In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, July, pp. 261–268.

Forne, C.J. and Hayes, M.P. (2001), 'A maximum likelihood approach to reconstructing scenes from photometric images', In *Proceedings of Image and Vision Computing New Zealand Conference, IVCNZ2001*, November, pp. 151–156.

Forne, C.J. and Hayes, M.P. (2002), 'Multiple camera scene reconstruction by belief propagation', In *Proceedings of Image and Vision Computing New Zealand Conference, IVCNZ2002*, November, pp. 309–314.

Forne, C.J. and Hayes, M.P. (2003), 'Scene reconstruction by greedy belief propagation', In *Proceedings of Image and Vision Computing New Zealand Conference, IVCNZ2003*, November, pp. 420–425.

Forne, C.J., Harding, C.M. and Lane, R.G. (2000), 'Optimal methods for multiple image matching', In *Proceedings of Image and Vision Computing New Zealand Conference, IVCNZ2000*, November, pp. 339–344.

Forney, G.D. (1973), 'The viterbi algorithm', In *Proceedings of the IEEE*, pp. 268–278.

Freedman, D. and Drineas, P. (2005), 'Energy minimization via graph cuts: Settling what is possible', In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE Computer Society, Washington, DC, USA, pp. 939–946.

Frey, B.J. and MacKay, D.J.C. (1998), 'A revolution: Belief propagation in graphs with cycles', In Jordan, M.I., Kearns, M.J. and Solla, S.A. (Eds.), *Advances in Neural Information Processing Systems*, The MIT Press.

Fua, P. and Leclerc, Y. (1995), 'Object-centered surface reconstruction: Combining multi-image stereo and shading', *International Journal of Computer Vision*, Vol. 16, No. 1, September, pp. 35–56.

Fusiello, A., Roberto, V. and Trucco, E. (1997), 'Efficient stereo with multiple windowing', In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June, pp. 858–863.

Gargallo, P. and Sturm, P. (2005), 'Bayesian 3D modeling from images using multiple depth maps', In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, California*, June, pp. 885–891.

Geman, S. and Geman, D. (1984), 'Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, November, pp. 721–741.

Gimel'farb, G. (1998), *Stereo Terrain Reconstruction by Dynamic Programming*, Technical Report CITR-TR-21, Center for Image Technology and Robotics, Computer Science Department, University of Auckland.

Gimel'farb, G.L. and Haralick, R.M. (1997), 'Terrain reconstruction from multiple views', In *Proceedings of the 7th International Conference on Computer Analysis of Images and Patterns*, pp. 694–701.

Gimel'farb, G.L. and Zhong, J.Q. (2001), 'Matching multiple views by the least square correlation', In *Proceedings of the 10th International Workshop on Theoretical Foundations of Computer Vision*, Springer-Verlag, London, UK, pp. 105–114.

Gong, M. and Yang, Y.H. (2001), 'Multi-resolution stereo matching using genetic algorithm', *IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV'01)*, pp. 21–29.

Greene, N. (1986), 'Environment mapping and other applications of world projections', *IEEE Computer Graphics and Applications*, Vol. 6, November, pp. 21–29.

Greig, D., Porteous, B. and Seheult, A. (1989), 'Exact maximum a posteriori estimation for binary images', *Journal of the Royal Statistical Society, Series B*, Vol. 51, No. 2, pp. 271–279.

Gruen, A.W. and Baltsavias, E.P. (1987), 'High-precision image matching for digital terrain model generation', *Photogrammetria*, Vol. 42, No. 3, December, pp. 97–112.

Gruen, A.W. and Baltsavias, E.P. (1988), 'Geometrically constrained multiphoto matching', *Photogrammetric Engineering and Remote Sensing*, Vol. 54, 5, pp. 633–641.

Guan, S. and Klette, R. (2008), 'Belief-propagation on edge images for stereo analysis of image sequences', In *Proceedings of Robot Vision 2008*, pp. 291–302.

Harding, C. (2001), *How Far Away is it? Depth estimation by a moving camera*, PhD thesis, Dept. of Electrical and Electronic Engineering, University of Canterbury, January.

Harding, C., Bainbridge-Smith, A. and Lane, R. (2000), 'Limits of tomographic depth estimation', In *Proceedings of SPIE — The international Society for Optical Engineering*, pp. 81–92.

Hastings, W.K. (1970), 'Monte carlo sampling methods using markov chains and their applications', *Biometrika*, Vol. 57, No. 1, pp. 97–109.

Healey, G.E. and Kondepudy, R. (1994), 'Radiometric CCD camera calibration and noise estimation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, March, pp. 267–276.

Henkel, R.D. (1997), 'Fast stereovision by coherence detection', In *Computer Analysis of Images and Patterns*, pp. 297–304.

Heskes, T. (2003), 'Stable fixed points of loopy belief propagation are local minima of the Bethe free energy', In Becker, S., Thrun, S. and Obermayer, K. (Eds.), *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, pp. 343–350.

Heskes, T. (2006), 'Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies', *Journal of Artificial Intelligence Research*, Vol. 26, pp. 153–190.

Heskes, T. and Zoeter, O. (2003), 'Generalized belief propagation for approximate inference in hybrid Bayesian networks', In Bishop, C.M. and Frey, B.J. (Eds.), *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*.

Hirschmuller, H. (2005), 'Accurate and efficient stereo processing by semi-global matching and mutual information', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, No. , pp. 807–814.

Hogbom, J.A. (1974), 'Aperture synthesis with a non-regular distribution of interferometer baselines', *Astronomy and Astrophysics Supplement Series*, Vol. 15, pp. 417–426.

Hong, L. and Chen, G. (2004), 'Segment-based stereo matching using graph cuts', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Vol. 1, pp. 74–81.

Ilic, S. and Fua, P. (2006), 'Radiometric CCD camera calibration and noise estimation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, February, pp. 328–333.

Intille, S. and Bobick, A. (1994), *Disparity-space images and large occlusion stereo*, Technical Report 220, Media Lab Perceptual Computing Group, M.I.T. Condensed version appears in ECCV, pp. 179-186, 1994.

Ishikawa, H. (2003), 'Exact optimization for markov random fields with convex priors', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 10, October, pp. 1333–1336.

Ishikawa, H. and Geiger, D. (1998), 'Occlusions, discontinuities, and epipolar lines in stereo', In *Proceedings of the 5th European Conference on Computer Vision*, pp. 89–96.

Jaynes, E.T. (2003), *Probability Theory: The Logic of Science*, Cambridge University Press.

Jensen, F.V. (2001), *Bayesian Networks and Decision Graphs*, Statistics for Engineering and Information Science, Springer Verlag.

Jonsson, E. and Felsberg, M. (2005), 'Efficient robust mean value computation of 1D features', In *Proceedings of the SSBA Symposium on Image Analysis*, Malmö, March.

Kalman, R.E. (1960), 'A new approach to linear filtering and prediction problems', *Transaction of the ASME Journal of Basic Engineering*, Vol. 82, March, pp. 34–45.

Kamberova, G. and Bajcsy, R. (1997), *The effect of radiometric correction on multicamera algorithms*, Technical Report MS-CIS-97-17, GRASP Lab, University of Pennsylvania.

Kanade, T. and Okutomi, M. (1994), 'A stereo matching algorithm with an adaptive window: Theory and experiment', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 9, September, pp. 920–932.

Kanade, T., Yoshida, A., Oda, K., Kano, H. and Tanaka, M. (1996), 'A stereo machine for video-rate dense depth mapping and its new applications', In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June.

Kang, S.B., Szeliski, R. and Chai, J. (2001), 'Handling occlusions in dense multi-view stereo', In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, pp. 103–110.

Kikuchi, R. (1951), 'A theory of cooperative phenomena', *Phys. Rev.*, Vol. 81, No. 6, March, p. 988.

Kim, J., Kolmogorov, V. and Zabih, R. (2003), 'Visual correspondence using energy minimization and mutual information', In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*, IEEE Computer Society, Washington, DC, USA, pp. 1033–1040.

Klaus, A., Sormann, M. and Karner, K. (2006), 'Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure', In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, pp. 15–18.

Kolmogorov, V. (2006), 'Convergent tree-reweighted message passing for energy minimization', *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 10, pp. 1568–1583.

Kolmogorov, V. and Rother, C. (2006), 'Comparison of energy minimization algorithms for highly connected graphs', In *ECCV (2)*, pp. 1–15.

Kolmogorov, V. and Zabih, R. (2001), 'Computing visual correspondence with occlusions using graph cuts', *Eighth International Conference on Computer Vision (ICCV'01)*, Vol. 2, p. 508.

Kolmogorov, V. and Zabih, R. (2002), 'Multi-camera scene reconstruction via graph cuts', In *Proceedings of the European Conference on Computer Vision (ECCV)*, May, pp. 82–96.

Kolmogorov, V. and Zabih, R. (2004), 'What energy functions can be minimized via graph cuts?', *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 2, pp. 147–159.

Kolmogorov, V., Zabih, R. and Gortler, S. (2003), 'Generalized multi-camera scene reconstruction using graph cuts', In *Proceedings of Energy Minimization Methods in Computer Vision and Pattern Recognition*, July, pp. 501–516.

Korb, K.B. and Nicholson, A.E. (2004), *Bayesian Artificial Intelligence*, Vol. 1 of Chapman and Hall/CRC Computer Science and Data Analysis, CRC Press.

Krupnik, A. (1996), 'Using theoretical intensity values as unknowns in multiple-patch least-squares matching', *Photogrammetric Engineering and Remote Sensing*, Vol. 62, No. 10, October, pp. 1151–1155.

Kschischang, F.R., Frey, B. and Loeliger, H..A. (2001), 'Factor graphs and the sum-product algorithm', *IEEE Transactions on Information Theory*, Vol. 47, No. 2, pp. 498–519.

Kutulakos, K.N. (2000), 'Approximate n-view stereo', In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part I*, Springer-Verlag, London, UK, pp. 67–83.

Kutulakos, K. and Seitz, S. (1998), *A Theory of Shape by Space Carving*, Technical Report 692, Computer Science Department, University of Rochester, May.

Lane, R. and Thacker, N. (1996), 'Stereo vision research: An algorithm survey'. http://citeseer.ist.psu.edu/24500.html.

Larsen, E.S., Mordohai, P., Pollefeys, M. and Fuchs, H. (2006), 'Simplified belief propagation for multiple view reconstruction', In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pp. 342–349.

Lee, S.H., Kanatsugu, Y. and Park, J.I. (2001), 'Hierarchical stochastic diffusion for disparity estimation', In *Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision. (SMBV 2001)*, December, pp. 111–120.

Lin, M.H. and Tomasi, C. (2004), 'Surfaces with occlusions from layered stereo', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 8, pp. 1073–1078.

Little, J.D., Murty, K.G., Sweeney, D.W. and Karel, C. (1963), 'An algorithm for the travelling salesman problem', *Operations Research*, Vol. 11, pp. 972–989.

Mansson, J. (1998), *Stereovision: A Model Of Human Stereopsis*, Technical Report, Lund University Cognitive Science, Technical Report.

Marr, D. and Poggio, T. (1976), 'Cooperative computation of stereo disparity', *Science*, Vol. 194, October, pp. 283–287.

McEliece, R.J., MacKay, D.J.C. and Cheng, J.F. (1998), 'Turbo decoding as an instance of pearl's 'belief propagation' algorithm', *IEEE Journal on Selected Areas in Communications*, Vol. 16, No. 2, pp. 140–152.

Meerbergen, G.V., Vergauwen, M., Pollefeys, M. and Gool, L.V. (2002), 'A hierarchical symmetric stereo algorithm using dynamic programming', *International Journal of Computer Vision*, Vol. 47, No. 1-3, pp. 275–285.

Meltzer, T., Yanover, C. and Weiss, Y. (2005), 'Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation', In *Proceedings of Tenth IEEE International Conference on Computer Vision (ICCV)*, IEEE Computer Society, October, pp. 428–435.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953), 'Equations of state calculations by fast computing machines', *Journal of Chemical Physics*, Vol. 21, No. 6, pp. 1087–1092.

Minka, T.P. (2001a), 'Expectation propagation for approximate Bayesian inference', In *UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 362–369.

Minka, T.P. (2001b), *A family of algorithms for approximate Bayesian inference*, PhD thesis, MIT Media Lab.

Mooij, J.M., Wemmenhove, B., Kappen, H.J. and Rizzo, T. (2007), 'Loop corrected belief propagation', In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS-07)*.

Murphy, K.P., Weiss, Y. and Jordan, M.I. (1999), 'Loopy belief propagation for approximate inference: An empirical study', In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 467–475.

Nicodemus, F.E., Richmond, J.C., Hsia, J.J., Ginsberg, I.W. and Limperis, T. (1977), *Geometrical Considerations and Nomenclature for Reflectance*, NBS Monograph 160, National Bureau of Standards, Washington, D.C., October.

Ohta, Y. and Kanade, T. (1985), 'Stereo by intra- and inter-scanline search using dynamic programming', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 2, March, pp. 139–154.

Okutomi, M. and Kanade, T. (1993), 'A multiple-baseline stereo', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 4, April, pp. 353–363.

Ouali, M., Lange, H. and Laurgeau, C. (1996), 'An energy minimization approach to dense stereovision', *Proceedings of the IEEE International Conference on Image Processing*, Vol. 2, September, pp. 841–845.

Paris, S., Sillion, F. and Quan, L. (2006), 'A surface reconstruction method using global graph cut optimization', *International Journal of Computer Vision*, Vol. 66, No. 2, February, pp. 141–161.

Park, J.I. and Inoue, S. (1998), 'Acquisition of sharp depth map from multiple cameras', *Signal Processing: Image Communication*, Vol. 14, No. 1-2, November 6,.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, California, revised second ed.

Pearl, J. (1996), 'Decision making under uncertainty', *ACM Comput. Surv.*, Vol. 28, No. 1, pp. 89–92.

POV-Ray (2007), 'Pov-ray – the persistence of vision raytracer'. A high-quality, totally free tool for creating stunning three-dimensional graphics. http://www.povray.org.

Preddey, J.T. and Lane, R.G. (1997), 'A tomographic technique for depth estimation from moving camera image sequences.', In *Proceedings of the Joint Australian and New Zealand Conference on Digital Image Computing, Techniques and Applications: DICTA-97*, pp. 23–28.

Rosenholm, D. (1987), 'Multi-point matching using the least-squares technique for evaluation of three-dimensional models', *Photogrammetric Engineering and Remote Sensing*, Vol. 53, No. 6, June, pp. 621–626.

Roy, S. and Cox, I.J. (1998), 'A maximum-flow formulation of the n-camera stereo correspondence problem', In *Proceedings of the IEEE International Conference on Computer Vision*, January.

Satoh, K. and Ohta, Y. (1996), 'Occlusion detectable stereo — systematic comparison of detection algorithms', In *Proceedings of the International Conference on Pattern Recognition*.

Scharstein, D. and Szeliski, R. (1996), 'Stereo matching with non-linear diffusion', In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June, pp. 343–350.

Scharstein, D. and Szeliski, R. (2002), 'A taxonomy and evaluation of dense two-frame stereo correspondence algorithms', *International Journal of Computer Vision*, Vol. 47, No. 1/2/3, pp. 7–42.

Seitz, S.M. and Dyer, C.R. (1999), 'Photorealistic scene reconstruction by voxel coloring', *International Journal of Computer Vision*, Vol. 35, No. 2, pp. 151–173.

Seitz, S.M., Curless, B., Diebel, J., Scharstein, D. and Szeliski, R. (2006), 'A comparison and evaluation of multi-view stereo reconstruction algorithms', In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, pp. 519–528.

Slabaugh, G., Malzbender, T. and Culbertson, B. (2000a), 'Volumetric warping for voxel coloring on an infinite domain', In *Proceedings of the Workshop on 3D Structure from Multiple Images for Large-scale Environments, (SMILE)*, pp. 41–50.

Slabaugh, G., Malzbender, T., Culbertson, B. and Schafer, R. (2000b), *Improved Voxel Coloring Via Volumetric Optimization*, Technical Report 3, Center for Signal and Image Processing, Georgia Institute of Technology.

Slabaugh, G., Culbertson, B., Malzbender, T. and Schafer, R. (2001), *A Survey of Volumetric Scene Reconstruction Methods from Photographs*, Technical Report 1, Center for Signal and Image Processing, Georgia Institute of Technology, February.

Snow, D., Viola, P. and Zabih, R. (2000), 'Exact voxel occupancy with graph cuts', In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June, pp. 345–352.

Strecha, C., Fransens, R. and Van Gool, L. (2004), 'Wide-baseline stereo from multiple views: A probabilistic account', In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, July, pp. 552–559.

Sun, C. (1997), 'A fast stereo matching method', In *Digital Image Computing: Techniques and Applications, Massey University, Auckland, New Zealand*, December, pp. 95–100.

Sun, C. (1999), 'Multi-resolution stereo matching using maximum-surface techniques', In *Proceedings of Digital Image Computing: Techniques and Applications, Perth, Australia*, December, pp. 195–200.

Sun, J., Shum, H.Y. and Zheng, N.N. (2002), 'Stereo matching using belief propagation', In *Proceedings of 7th European Conference of Computer Vision, Part II*, May, pp. 510–524.

Sun, J., Zheng, N.N. and Shum, H.Y. (2003), 'Stereo matching using belief propagation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 7, July, pp. 787–800.

Sun, J., Li, Y., Kang, S.B. and Shum, H.Y. (2005), 'Symmetric stereo matching for occlusion handling', In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June, pp. 399–406.

Szeliski, R. and Golland, P. (1999), 'Stereo matching with transparency and matting', *International Journal of Computer Vision*, Vol. 32, No. 1, pp. 45–61.

Szeliski, R. and Scharstein, D. (2002), 'Symmetric sub-pixel stereo matching', In *Proceedings of the 7th European Conference on Computer Vision-Part II ECCV*, Springer-Verlag, London, UK, pp. 525–540.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M.F. and Rother, C. (2006), 'A comparative study of energy minimization methods for markov random fields.', In *ECCV (2)*, pp. 16–29.

Tang, B., Ait-Boudaoud, D., Matuszewski, B.J. and kwan Shark, L. (2006), 'An efficient feature based matching algorithm for stereo images', *Geometric Modeling and Imaging — New Trends, 2006*, Vol. 0, pp. 195–202.

Tanner, R.M. (1981), 'A recursive approach to low complexity codes', *IEEE Transactions on Information Theory*, Vol. 27, No. 5, September, pp. 533–547.

Tappen, M.F. and Freeman, W.T. (2003), 'Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters', In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*, pp. 900–907.

Tran, S. and Davis, L. (2006), '3D surface reconstruction using graph cuts with surface constraints.', In *ECCV (2)*, pp. 219–231.

Tsai, R.Y. (1987), 'A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses', *IEEE Journal of Robotics and Automation*, Vol. 3, No. 4, August, pp. 323–344.

Tsin, Y., Kang, S. and Szeliski, R. (2003), 'Stereo matching with reflections and translucency', In *Conference on Computer Vision and Pattern Recognition*, pp. 702–709.

Veksler, O. (2002), 'Dense features for semi-dense stereo correspondence', *International Journal of Computer Vision*, Vol. 47, No. 1-3, pp. 247–260.

Vogiatzis, G., Torr, P. and Cippola, R. (2005), 'Multi-view stereo via volumetric graph-cuts', In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR-2005)*, pp. 391–399.

Wainwright, M.J., Jaakkola, T.S. and Willsky, A.S. (2005), 'MAP estimation via agreement on trees: Message-passing and linear programming', *IEEE Transactions on Information Theory*, Vol. 51, November, pp. 3697–3717.

Weinshall, D. (1991), 'Seeing ghost planes in stereo vision', *Vision Research*, Vol. 31, No. 10, pp. 1731–1748.

Weinshall, D. (1993), 'The computation of multiple matching in doubly ambiguous stereograms with transparent planes', *Spatial Vision*, Vol. 7, No. 2, pp. 183–198.

Weiss, Y. and Freeman, W.T. (2001), 'On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs', *IEEE Transactions on Information Theory*, Vol. 47, No. 2, February, pp. 736–744.

Yang, Q., Wang, L., Yang, R., Stewenius, H. and Nistér, D. (2006), 'Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling', In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, pp. 2347–2354.

Yazdani, M.R., Hemati, S. and Banihashemi, A.H. (2004), 'Improving belief propagation on graphs with cycles', *IEEE Communications Letters*, Vol. 8, No. 1, January, pp. 57–59.

Yedidia, J., Freeman, W. and Weiss, Y. (2002a), *Understanding Belief Propagation and its Generalizations*, Technical Report MERL-TR-2001-22, Mitsubishi Electric Research Laboratory, January.

Yedidia, J., Freeman, W. and Weiss, Y. (2002b), *Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms*, Technical Report MERL-TR-2002-35, Mitsubishi Electric Research Laboratory, August.

Yuille, A.L. (2002), 'CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation', *Neural Computation*, Vol. 14, No. 7, pp. 1691–1722.

Zitnick, C.L. and Kanade, T. (1999), *A Cooperative Algorithm for Stereo Matching and Occlusion Detection*, Technical Report CMU-RI-TR-99-35, The Robotics Institute, Carnegie Mellon University.

Zitnick, C.L. and Kang, S.B. (2007), 'Stereo for image-based rendering using image over-segmentation', *International Journal of Computer Vision*, Vol. 75, No. 1, pp. 49–65.